

**Arquitectura Multi-agente para JITIK - Un Sistema  
de Flujo de Información y Conocimiento**



**TESIS**

**Maestría en Ciencias en Sistemas Inteligentes**

**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY**

**POR**

**Erika de los Angeles Buenrostro Cruz**

**Mayo 2008**

**Arquitectura Multi-agente para JITIK - Un sistema  
de Flujo de Información y Conocimiento**



**T E S I S**

**Maestría en Ciencias en Sistemas Inteligentes**

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Por**

**Erika de los Angeles Buenrostro Cruz**

**Mayo 2008**

# **Arquitectura Multi-agente para JITIK - Un sistema de Flujo de Información y Conocimiento**

por

**Erika de los Angeles Buenrostro Cruz**

**Tesis**

Presentada al Programa de Graduados de la  
Escuela de Tecnologías de Información y Electrónica  
como requisito parcial para obtener el grado académico de

**Maestro en Ciencias**

especialidad en

**Sistemas Inteligentes**

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Campus Monterrey**

Mayo de 2008

Dedico el presente trabajo a mis padres y a la memoria de mi hermana Selene (†).

## Reconocimientos

Deseo externar un sincero agradecimiento a las personas que de alguna forma colaboraron en el desarrollo de esta tesis.

Al Dr. José Luis Aguirre, asesor de este trabajo de investigación y de quien he recibido gran apoyo para la conclusión de este proyecto, y a mis sinodales, el Dr. Ramón Brena y el Dr. Miguel Ángel Pérez.

ERIKA DE LOS ANGELES BUENOSTRO CRUZ

*Instituto Tecnológico y de Estudios Superiores de Monterrey*  
*Mayo 2008*

# Arquitectura Multi-agente para JITIK - Un sistema de Flujo de Información y Conocimiento

Erika de los Angeles Buenrostro Cruz, M.C.  
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2008

Asesor de la tesis: Dr. José Luis Aguirre Cervantes

El documento presenta el trabajo de investigación realizado para obtener el grado de Maestría en Ciencias con especialidad en Sistemas Inteligentes.

La tesis parte de la idea de que de los procesos básicos de la Administración del Conocimiento: desarrollo de nuevo conocimiento, asegurar el nuevo y el existente conocimiento, distribución del conocimiento y combinación del conocimiento [5]; es sin duda la distribución de conocimiento la que representa un mayor reto para las organizaciones debido principalmente a la velocidad con la que se genera en la actualidad y a la diversidad de fuentes donde se almacena.

Para ayudar a solventar esa situación se diseña la arquitectura de JITIK (Just In Time Information and Knowledge) basada en sistemas multi-agentes con el objetivo de permitir la interacción de diversos repositorios de información y conocimiento que ayude en los servicios de distribución de conocimiento.

Para lo anterior se presenta primero un conjunto de fuentes donde se encuentran almacenados la información y el conocimiento computacionalmente. Posteriormente se diseña la arquitectura multi-agentes definiendo roles y servicios que exploten esas fuentes. Posteriormente, se presenta un prototipo de la arquitectura de JITIK que incluye los roles de agentes y algunos servicios de distribución, para finalmente mostrar escenarios de prueba que validen el funcionamiento de la arquitectura.

# Índice general

<b>Reconocimientos</b>	<b>v</b>
<b>Resumen</b>	<b>vi</b>
<b>Índice de cuadros</b>	<b>x</b>
<b>Índice de figuras</b>	<b>xi</b>
<b>Capítulo 1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	1
1.2. Definición del Problema . . . . .	3
1.3. Objetivos . . . . .	5
1.3.1. Objetivos Generales . . . . .	5
1.3.2. Particulares . . . . .	5
1.4. Hipótesis . . . . .	5
1.5. Organización de la Tesis . . . . .	6
<b>Capítulo 2. Marco Teórico</b>	<b>7</b>
2.1. Administración del Conocimiento . . . . .	7
2.2. Tecnologías de Apoyo para la Administración del Conocimiento . . . . .	12
2.3. Sistemas Multi-agentes . . . . .	14
2.3.1. FIPA-ACL . . . . .	16
2.3.2. Ontología . . . . .	19
2.3.3. Herramientas de Implementación de Sistemas Multi-agentes . . . . .	19
2.4. Proyecto RICA (antes CORREA) . . . . .	24
<b>Capítulo 3. Arquitectura multi-agentes para JITIK</b>	<b>28</b>
3.1. Descripción General de Sistema JITIK . . . . .	29
3.2. Arquitectura multi-agente de JITIK . . . . .	32
3.3. Roles de Agentes . . . . .	34
3.4. Servicios de JITIK . . . . .	38

<b>Capítulo 4. Prototipo y pruebas</b>	<b>44</b>
4.1. Prototipo de JITIK . . . . .	45
4.1.1. Roles de agentes en el prototipo de JITIK . . . . .	45
4.2. Servicios de JITIK . . . . .	52
4.3. Pruebas . . . . .	62
4.3.1. ESCENARIO DE PRUEBA A - Aviso de noticias relacionadas con la competencia . . . . .	63
4.3.2. ESCENARIO DE PRUEBA B - Coleccionista de modelos de au- tos a escala . . . . .	66
4.3.3. ESCENARIO DE PRUEBA C - Nuevos libros en biblioteca . . . . .	68
<b>Capítulo 5. Conclusiones</b>	<b>70</b>
5.1. Conclusiones . . . . .	70
5.2. Aportaciones . . . . .	71
5.3. Trabajos relacionados . . . . .	72
5.4. Trabajos futuros . . . . .	77
5.4.1. Nuevos servicios . . . . .	77
5.4.2. Mejoras a la arquitectura de JITIK . . . . .	78
5.4.3. Relacionados con la Administración del Conocimiento . . . . .	79
<b>Capítulo 6. Apéndice A - Pseudocódigo del prototipo de JITIK</b>	<b>80</b>
6.1. Pseudocódigo del agente de sitio . . . . .	80
6.2. Pseudocódigo del agente de ontología . . . . .	81
6.3. Pseudocódigo del agente personal . . . . .	82
6.4. Pseudocódigo del agente monitor . . . . .	82
6.5. Pseudocódigo del agente bridge o puente . . . . .	83
<b>Capítulo 7. Apéndice B - Tablas de datos para prototipo JITIK</b>	<b>84</b>
7.1. Tabla Usuarios . . . . .	84
7.2. Tabla PáginasWeb . . . . .	84
7.3. Tabla Grupos . . . . .	85
7.4. Tabla Areas . . . . .	85
7.5. Tabla LibrosxArea . . . . .	85
7.6. PaginasWebxUsuario . . . . .	86
7.7. Tabla GruposxUsuario . . . . .	86
7.8. Tabla AreasxUsuario . . . . .	86
<b>Capítulo 8. Apéndice C - Taxonomía de ACM para la Inteligencia Arti- ficial</b>	<b>87</b>
<b>Bibliografía</b>	<b>90</b>





## Índice de cuadros

2.1. Atributos de un agente . . . . .	15
2.2. Performativas de FIPA-ACL clasificados por función . . . . .	18
3.1. Rol del agente de sitio . . . . .	35
3.2. Rol del agente de ontología . . . . .	35
3.3. Rol del agente personal . . . . .	36
3.4. Rol del agente monitor . . . . .	37
3.5. Rol del agente bridge o puente . . . . .	37
4.1. Rol del agente Agent Management Systems de JADE . . . . .	47
4.2. Rol del agente Directory Facilitator de JADE . . . . .	47
4.3. Instancia del agente de sitio para el prototipo de JITIK . . . . .	48
4.4. Instancia del agente de ontología para el prototipo de JITIK . . . . .	49
4.5. Instancia del agente personal para el prototipo de JITIK . . . . .	49
4.6. Instancia del agente monitor en el prototipo de JITIK . . . . .	50
4.7. Instancia del agente bridge o puente para el prototipop de JITIK . . . . .	51
4.8. Sitios Web utilizados para prueba de monitoreo . . . . .	63
5.1. Trabajos relacionados de agentes inteligentes con AC . . . . .	73

## Índice de figuras

2.1. Procesos básicos del conocimiento . . . . .	8
2.2. Arquitectura distribuida de JADE . . . . .	20
2.3. Arquitectura de comunicación de JADE . . . . .	21
2.4. Arquitectura interna de JADE . . . . .	21
2.5. Componentes de ZEUS . . . . .	22
2.6. Estructura de ZEUS . . . . .	23
2.7. Arquitectura de RICA . . . . .	26
2.8. Componentes de de RICA . . . . .	27
3.1. Arquitectura conceptual de JITIK . . . . .	28
3.2. Arquitectura distribuida de JITIK . . . . .	32
3.3. Arquitectura de JITIK . . . . .	34
3.4. Diagrama UML para el servicio de páginas blancas . . . . .	38
3.5. Diagrama UML para el servicio de páginas amarillas . . . . .	39
3.6. Diagrama UML para el servicio de ontología . . . . .	40
3.7. Diagrama UML para el servicio de mensajería . . . . .	41
3.8. Diagrama UML para el servicio de monitoreo . . . . .	42
3.9. Diagrama UML para el servicio de bridge o puente . . . . .	43
4.1. Arquitectura del prototipo de JITIK . . . . .	45
4.2. Diagrama de base de datos para el prototipo de JITIK . . . . .	46
4.3. Arquitectura de JITIK integrada con plataforma de JADE . . . . .	51
4.4. Secuencia de pasos en el servicio de mensajería por grupo . . . . .	53
4.5. Diagrama UML para el servicio de mensajería por grupos . . . . .	54
4.6. Secuencia de pasos en el servicio de mensajería por áreas . . . . .	55
4.7. Diagrama UML para el servicio de mensajería por área de interés . . . . .	56
4.8. Secuencia de pasos en el envío de mensajería por usuario . . . . .	57
4.9. Diagrama UML para el servicio de mensajería por usuario . . . . .	58
4.10. Secuencia de pasos para el servicio de monitoreo de páginas Web . . . . .	59
4.11. Diagrama UML para el servicio de monitoreo de páginas Web . . . . .	60
4.12. Secuencia de pasos en el servicio de bridge con el sistema de biblioteca . . . . .	61
4.13. Diagrama UML para el servicio bridge con el sistema externo de biblioteca . . . . .	62

4.14. Sitio de noticias de WebSense . . . . .	65
4.15. Sitio de noticias de TrendMicro . . . . .	65
4.16. Sitio de noticias de BlueCoat . . . . .	65
4.17. Auto minuatara en subasta en eBay . . . . .	67
4.18. Libro nuevo en biblioteca digital del ITESM . . . . .	69
5.1. Comparación entre arquitecturas de RICA y JITIK . . . . .	72

## Capítulo 1

### Introducción

#### 1.1. Antecedentes

El campo de la Administración del Conocimiento ha generado gran revuelo en los últimos años. El número de conferencias sobre el tema, libros y revistas, alrededor del mundo, indican el grado de interés que ha despertado en las organizaciones.

“Hasta hace poco se hablaba de la era industrial, pero hoy los negocios se rigen por el concepto de sociedad del conocimiento o de la información” [7] . El Internet y la rapidez para compartir información entre las organizaciones hacen del conocimiento el principal activo para hacer negocios de forma eficiente, así que la administración del conocimiento se vuelve un tema crucial para las organizaciones que desean ser exitosas.

Organizaciones de todo tipo se han dado cuenta que el conocimiento es su principal ventaja competitiva. Conforme el conocimiento se convierte en el recurso estratégico del futuro, la necesidad de las organizaciones por desarrollar un entendimiento de las estrategias de conocimiento, procesos y herramientas para la creación, transferencia y desarrollo de esta ventaja única se vuelve crítico.[2]

El gran auge que ha tenido la Administración de Conocimiento en el ámbito de negocios, exige que se hagan esfuerzos computacionales que soporten el concepto. Además el desarrollo de internet ha proporcionado la disponibilidad de tecnología a bajo costo para el rápido despliegue de la infraestructura tecnológica requerida para la recopilación de conocimiento en los negocios y su posterior distribución a aquellos que puedan obtener ventajas competitivas de él.

La tarea de aplicar Administración del Conocimiento como nueva disciplina dentro de una organización es un gran esfuerzo. Sin embargo, para negocios que deben competir diariamente en un mundo cambiante, una Administración del Conocimiento efectiva es la clave para la innovación, productividad y crecimiento.[2]

Una de las principales barreras para que una iniciativa de conocimiento sea exitosa es distinguir entre conocimiento e información. Información consiste de gran cantidad de datos organizados, agrupados y categorizados en patrones que tienen un significado; Conocimiento es información que se utiliza con el fin de ejecutar la acción correcta. Administrar conocimiento es por lo tanto cualitativamente diferente de administrar información.[13]

La información se convierte en conocimiento a través de un proceso humano de compartir entendimiento y hacer que tenga sentido tanto a nivel personal como organizacional. La Administración del Conocimiento comienza enfatizando la importancia de la gente, sus prácticas de trabajo y su cultura de trabajo antes de decidir donde y qué tecnología integrar. La Administración de Información, por otro lado, frecuentemente comienza con una solución tecnológica primero con consideraciones de prácticas de trabajo y cultura de trabajo en segundo plano. [13]

Como apoyo a la administración del conocimiento surge el concepto de Redes de Conocimiento, cuyo principal objetivo es mejorar los procesos de negocios de una organización y ayudar a éstas a competir más efectivamente. [5] Las Redes de Conocimiento son relaciones entre entidades (individuos, equipos, organizaciones) trabajando en un objetivo común para crear y compartir conocimiento.[2]

Una Red de Conocimiento puede apoyarse en varias tecnologías computacionales para lograr sus objetivos; denominaremos a dichas redes como Redes Informáticas de Conocimiento (RIC). Entre las tecnologías en que se puede basar la construcción de una RIC, y más en particular los aspectos relacionados con la localización y la “compartición” de conocimiento, podemos señalar, entre otras, las bibliotecas digitales, los servicios Web, las páginas Web y las memorias organizacionales. Mención aparte merecen las tecnologías de agentes inteligentes que pueden realizar varias de las actividades necesarias para la implementación de una RIC, y que son objeto de la tesis que aquí se presenta.

La presente tesis plantea la posibilidad de que un sistema basado en tecnología de agentes inteligentes, específicamente el sistema JITIK (Just In Time Information and Knowledge), tome lugar en una arquitectura multi-agente que de soporte a algunos los servicios de la administración del conocimiento.

La primera iniciativa para este proyecto fue en 1998 con el proyecto CORREA (COoRdinación de Recursos de Educación e investigación mediante Agentes) que nació de la necesidad de integración que tiene el ITESM entre sus campus, para el mejor aprove-

chamamiento de los recursos materiales y humanos. El objetivo era cubrir la necesidad de comunicación con otras instituciones mexicanas principalmente las relacionadas con investigación en computación. El proyecto CORREA se desarrolló en el Centro de Inteligencia Artificial del ITESM con apoyo de la Red de Investigación en Informática del CONACYT.[14]

Para el desarrollo del proyecto CORREA se utilizó la tecnología de los Sistemas Basados en Agentes para llevar a la práctica un servicio de apoyo a la coordinación de los grupos de investigación en computación. Dicho apoyo se traduce concretamente en funciones como las siguientes:

- Direccionar mensajes a grupos de investigadores, según sus áreas de interés.
- Monitorear grupos de noticias de netnews, enviando notificaciones a investigadores.
- Monitorear bases de datos sobre reportes de investigación y otros recursos, notificando automáticamente sobre cambios ocurridos en aquellas.
- Responder a interrogaciones sobre qué investigadores participan en qué áreas de interés, o bien cuál es la dirección electrónica de algún investigador.

En 1999, se le cambió el nombre a RICA (Redes de Información y Conocimiento mediante Agentes) para enfatizar la utilización de la tecnología de agentes en la interconexión de la red de conocimiento. A pesar de que en los proyectos de CORREA y RICA incluía los conceptos de sistemas basados en agentes, el modelo de la arquitectura era básico como se explicará en el marco teórico más adelante. Además, las implementaciones no utilizaban la tecnología de agentes inteligentes.

En el año 2001, el proyecto toma el nombre de JITIK (Just In Time Information and Knowledge) con el fin de hacer incapié en la importancia que tiene acercar la información y el conocimiento a la persona adecuadas en el momento preciso.

## 1.2. Definición del Problema

Dentro de los procesos básicos del conocimiento: desarrollo de nuevo conocimiento, asegurar el nuevo y el existente conocimiento, distribución del conocimiento y combinación del conocimiento; que se describen a detalle más adelante, es sin duda la distribución de conocimiento la que representa un mayor reto para las organizaciones.[5]

Las organizaciones en la actualidad tienen al alcance múltiples formas de hacer negocios ya que las barreras de comunicación y transporte se han minimizado. Una

empresa puede vender sus productos al otro lado del mundo o tener una tienda en línea 24 horas, 365 días del año. Este nuevo entorno definido como global, en cuanto a que las organizaciones y empresas pueden realizar actividades prácticamente en cualquier parte del mundo, ha sido construido en gran medida gracias a la Tecnologías de Información. Herramientas como el Internet, correo electrónico, teléfono celular y la eficiencia en el transporte hace que las transacciones y el intercambio de información sean casi instantáneas.

Lo anterior acarrea mayor exigencia en cuestión de competitividad para las organizaciones, pues ahora tienen que competir en un entorno global. Ser más eficientes y contar con el conocimiento adecuado, en el lugar adecuado y en el momento adecuado se vuelve crítico para la toma de decisiones. Por lo que el flujo o distribución de conocimiento dentro de una organización puede resultar fundamental para su éxito.

Enseguida se listan los principales retos que conlleva una adecuada distribución del conocimiento:

- El conocimiento se genera en algún punto en la organización pero no está disponible de forma inmediata en otro punto donde puede utilizarse.
- La distribución geográfica de las organizaciones hace que exista retraso en el intercambio de conocimiento.
- El conocimiento se duplica.
- Se requiere tiempo y esfuerzo para ir a buscar el conocimiento.
- El conocimiento cambia constantemente y requiere ser actualizado.

Sin la distribución o flujo de conocimiento, el conocimiento se quedaría estático. Es el flujo de conocimiento entre individuos que permite que se perfeccione el conocimiento existente o se genere nuevo conocimiento.

Como se mencionó anteriormente las Tecnologías de Información tienen mucho que aportar en lo que a distribución del conocimiento se refiere. Particularmente las Tecnologías de Sistemas Multi-Agentes brindan una base tecnológica más sólida que las tecnologías convencionales para crear una Red Informática del Conocimiento que permita solventar los problemas relacionados con la distribución del conocimiento.

El problema que se estudia en esta tesis tiene que ver precisamente con el diseño de una arquitectura multi-agentes y la implementación de un prototipo de software para comprobar este hecho.



## 1.3. Objetivos

### 1.3.1. Objetivos Generales

Validar que una arquitectura de sistemas multi-agentes que permita la interacción de diversos repositorios de conocimiento puede ayudar en los servicios de distribución de conocimiento.

### 1.3.2. Particulares

- Investigar los diversos repositorios de conocimiento que existen.
- Diseñar una arquitectura de software basada en sistemas multi-agentes para proporcionar servicios de distribución de conocimiento.
- Diseñar los roles de los agentes para implementar funciones de localización y distribución del conocimiento.
- Investigar herramientas adecuadas para el desarrollo del prototipo.
- Implementar un prototipo de la arquitectura diseñada con los servicios y roles de los agentes.
- Diseñar escenarios de prueba.

## 1.4. Hipótesis

Es posible que las arquitecturas multi-agentes puedan ser utilizadas en la construcción de sistemas para soportar el proceso de distribución de conocimiento ofreciendo una mejor alternativa que mecanismos de software tradicional.

Las preguntas que deben plantearse durante esta investigación para alcanzar los objetivos planteados son:

- ¿En donde se encuentra contenido el conocimiento computacionalmente?
- ¿Qué características debe tener una arquitectura de software que integre sistemas con tecnología de agentes inteligentes, para una plataforma que ofrezcan principalmente el servicio de distribución del conocimiento?
- ¿Qué herramientas computacionales existen para diseñar un prototipo?
- ¿Qué características debe tener un software que permita integrar estas tecnologías?

- ¿Qué componentes y servicios debe tener el sistema JITIK para permitir la distribución de conocimiento?

Durante la tesis se propone utilizar el proyecto JITIK como un medio para validar los elementos que la investigación vaya arrojando. Lo anterior no significa que los estudios en cuestión deban ser limitados a JITIK, es deseable que el resultado de la investigación sea aplicable a otras iniciativas de implementación para brindar soporte a sistemas de administración del conocimiento. [14]

## 1.5. Organización de la Tesis

A continuación se describe de manera general el contenido del resto del documento de tesis:

En el Capítulo II - Marco Teórico, se presenta el estado del arte los conceptos relacionados con el presente trabajo de investigación como son: la Administración del Conocimiento, las tecnologías de apoyo a la Administración del Conocimiento así como los Sistemas Multi-agentes.

Enseguida el Capítulo III - Arquitectura multi-agentes para JITIK, se muestra la arquitectura de JITIK basada en la teoría de sistemas multi-agentes y aplicada a la distribución del conocimiento.

A continuación en el Capítulo IV - Prototipo de JITIK y pruebas, se presenta la implementación del prototipo del JITIK, los ejemplos de servicios implementados y las pruebas realizadas.

Finalmente en el Capítulo V - Conclusiones se mencionan las conclusiones obtenidas en el presente trabajo de tesis así como las propuestas de extensión y trabajos futuros.

## Capítulo 2

### Marco Teórico

En este capítulo se detallan los conceptos relacionados con este trabajo de tesis, la Administración del Conocimiento orientándose más al tema de distribución del conocimiento y apoyándose en conceptos como Redes del Conocimiento y Memoria Organizacional. Se habla también de Tecnologías de Información que sirven de apoyo a los procesos básicos de la Administración del Conocimiento, así como de los Sistemas Multi-agentes y sus características principales. Finalmente se presentan una serie de herramientas disponibles en Internet y que se utilizan para la implementación de Sistemas Multi-agentes.

#### 2.1. Administración del Conocimiento

Administración del Conocimiento es el proceso de crear valor de los activos intangibles de una organización. Activos intangibles, también conocidos como capital intelectual.[8]

La administración del conocimiento, según Karl M. Wiig, tiene como objetivos:

1. Hacer que una empresa actúe tan inteligente como sea posible para asegurar la viabilidad y sobre todo el éxito y,
2. realizar la mejor valuación de su capital intelectual.

Administración del Conocimiento también puede ser definida como un conjunto de herramientas para la automatización de relaciones deductivas o inherentes entre objetos de información, usuarios y procesos. [4]

Las organizaciones están adoptando la Administración del Conocimiento por varias razones. La razón principal es incrementar la innovación, otro factor importante es que involucrarse en administración de conocimiento significa retención del conocimiento, retención de la gente y eso repercute en la visión. Capturando el conocimiento clave

de los expertos antes de que se retiren, la retención de conocimiento puede ser incrementado creando una Memoria Organizacional o base de conocimiento basado en Administración de Conocimiento. [8]

El concepto no es nuevo, siempre se ha sabido que el conocimiento es necesario para la solución eficiente de los problemas. Sin embargo, es hasta ahora que ha tomado especial atención el hecho de que el conocimiento que atesoran las empresas, puede ser explotado para tomar ventajas competitivas.[5]

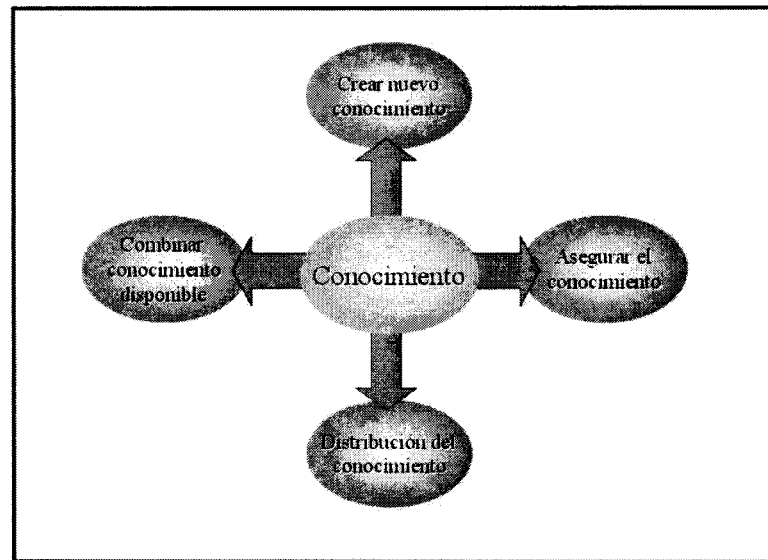


Figura 2.1: Procesos básicos del conocimiento

Según los estudiosos, se distinguen cuatro procesos básicos del conocimiento, los cuales se muestran en la Figura 2.1.

1. *Desarrollo de nuevo conocimiento.* Las compañías sobreviven gracias al desarrollo de nuevo conocimiento basado en ideas creativas, el análisis de errores, la experiencia diaria y un trabajo fuerte en los departamentos de investigación y desarrollo.
2. *Asegurar el existente y el nuevo conocimiento.* El conocimiento debe hacerse accesible para otros en las organizaciones que necesiten de ese conocimiento. Este conocimiento debe estar disponible en el lugar y tiempo correcto.
3. *Distribución del conocimiento.* El conocimiento debe ser activamente distribuido a aquellos que puedan hacer uso de él. La velocidad con la que circule el conocimiento en una compañía es crucial para los procesos del negocio. La distribución

del conocimiento puede conseguirse por varios métodos - por transferencia de personas, organizando cursos, dando presentaciones o a través de video-noticias internos.

4. *Combinación de conocimiento.* Una compañía puede desempeñarse de la mejor manera, sólo si todo el conocimiento disponible de las áreas es combinado. Los productos y servicios cada vez más son desarrollados por equipos multidisciplinarios.

Una buena Administración del Conocimiento implica la continua alineación de estos procesos para mejorar la capacidad de aprendizaje de una organización.[1]

Cada uno de los procesos anteriores, pueden en determinado momento hacer fracasar la implementación de un sistema de Administración del Conocimiento. Sin embargo, la distribución del conocimiento en particular es muy importante para las organizaciones, ya que frecuentemente se genera conocimiento importante en una parte de la organización, pero no es utilizado en otra con oportunidad.

La distribución del conocimiento, pensada en procedimientos tradicionales como entrevistas, cursos, conferencias, etc. resulta muy dependiente a factores organizacionales e incluso culturales, entre los cuales podemos mencionar, disponibilidad de los poseedores del conocimiento, empatía entre el que tiene el conocimiento y el que desea adquirirlo, burocracia, prejuicios, exceso de niveles, falta de motivación, entre otros.

Desde esa perspectiva, la distribución del conocimiento y de la información es un área donde la tecnología puede resultar de gran ayuda. El desarrollo de Internet ha proporcionado la disponibilidad de tecnología a bajo costo para el rápido despliegue de la infraestructura requerida para la recopilación de conocimiento relevante al negocio y su posterior distribución a aquellos que puedan obtener ventajas competitivas de él.

Las empresas grandes y distribuidas son las que más pueden beneficiarse de la aplicación de la tecnología de información en distribución del conocimiento.

Existen dos tipos de conocimiento, el Conocimiento Explícito que es el tipo de conocimiento que puede ser empaquetado como información, se encuentra en reportes, imágenes, manuales, video, etc. Y el Conocimiento Tácito que es el conocimiento que tienen los individuos como experiencia personal que se intercambia mediante un contacto directo. [1]

Por ejemplo, el reporte del Plan Estratégico del Negocio es algo que puede circular por la organización y los empleados puede leer y ejecutar el plan, eso sería Conocimien-

to Explícito. Por otro lado, el Conocimiento Tácito sería el proceso de decisión que sigue un Gerente para resolver cierto problema, el conocimiento de cómo resolver el problema es la habilidad y capacidad del Gerente. [11]

Aunque es más difícil de administrar el Conocimiento Tácito es el más relevante para la Administración del Conocimiento ya que es donde reside el valor estratégico para una organización.[4]

El flujo de conocimiento es el objetivo principal de la Administración del Conocimiento ya que es el que soporta la interacción entre el Conocimiento Tácito que es intercambiado y generado en comunidades de trabajadores de conocimiento, con el Conocimiento Explícito que es el que se encuentra en los repositorios de conocimiento en forma de bibliotecas o archivos de documentos.[1]

Existen otros conceptos relacionados con la Administración del conocimiento y que a continuación se describen debido a la relevancia que tienen para efectos de la tesis.

#### *Redes Informáticas del Conocimiento*

Las Redes de Conocimiento tienen cinco características importantes que las diferencian de otras estructuras organizacionales similares (Wenger, 1999, Wenger and Zin-der, 2000). Estas características son:

- son responsables de crear, compartir, proteger y mejorar el conocimiento;
- son redes de trabajo orientadas a un propósito;
- requieren de un compromiso organizacional más allá del compromiso de los participantes;
- son construidas en base a experiencia, no sólo un interés o práctica; y
- están orientadas al desarrollo y fortalecimiento de las capacidades de aprendizaje de todos los miembros. [2]

#### *Memoria Organizacional*

“Una memoria corporativa es una representación explícita, disímbola y persistente del conocimiento e información en una organización” [1] Toda pieza de información o conocimiento que contribuya en el desempeño de la organización debe ser almacenado en una memoria organizacional.

El objetivo final de una organización no es maximizar el desarrollo de conocimiento, almacenamiento distribución o combinación por sí mismo, sino para mejorar

su poder competitivo adaptando continuamente la organización a ambientes externos (mercado, clima político y social, preferencia de clientes, etc.) Los requerimientos de implementación de una memoria corporativa deben por lo tanto ser presentados en términos de este proceso de adaptación, léase aprendizaje. [1]

Según Borghoff y Pareschi (1998), una memoria organizacional es una representación explícita y persistente de la información de una organización, y su principal función es mejorar la competitividad de la misma. El objetivo principal de la memoria organizacional es recordar, por lo que se realizan cuatro procesos básicos:

- *Selección.* El proceso de selección evalúa la entrada del conocimiento la cual se realiza con base en la estrategia del negocio: lo que la empresa define como importante para lograr sus objetivos. La memoria organizacional sólo debe contener información que esté en posibilidades de generar valor para la empresa.
- *Codificación.* Para que la información captada para una empresa pueda ser almacenada, primero debe codificarse. La codificación implica que la empresa debe transformar su conocimiento para que sea almacenado y entendible por cualquier persona que lo recupere, aunque la información almacenada puede ser recuperada exitosamente sólo si quien la accede tiene la capacidad de decodificarla.
- *Almacenamiento.* La etapa de almacenamiento se encarga de tomar las unidades de información ya codificadas, las clasifica y las guarda en la memoria organizacional. El conocimiento debe estar organizado y representado en diferentes estructuras de conocimiento para poder ser almacenado. La parte medular está en la clasificación de la información.
- *Recuperación.* Una vez codificada y almacenada la información en los repositorios, el último proceso es la recuperación. En esta etapa se extrae la información de los distintos repositorios para su uso.[9]

Existen tecnologías que pueden apoyar en cada uno de estos procesos básicos. Para el proceso de Selección, existen robots, agentes inteligentes, servicios Web, plataformas de trabajo colaborativo, por mencionar algunas que pueden ayudar en la selección del conocimiento relevante para los objetivos de una organización.

En el caso de la codificación y almacenamiento, tecnologías de bases de datos, estructuras de datos, documentos XML, repositorios de objetos, etc. Pueden apoyar para guardar de forma efectiva el conocimiento para que posteriormente sea extraído.

Finalmente, en el proceso de recuperación tecnologías como servicios Web, queries, búsquedas, pueden servir para recuperar el conocimiento y así utilizarlo cuando sea conveniente.

## 2.2. Tecnologías de Apoyo para la Administración del Conocimiento

En lo que se refiere a Administración del Conocimiento existen varias tecnologías que pueden apoyar en los procesos básicos:

### Desarrollo de nuevo conocimiento

Se trata de generar conocimiento a partir del conocimiento existente. Las tecnologías que pueden aportar son:

#### *Mesa de ayuda*

Es un sistema de información donde se registran los problemas que se están generando en una organización y un grupo de trabajo busca una solución en conjunto. La resolución de problemas queda documentada y hace que ese conocimiento pueda ser reutilizado cuando un problema similar vuelva a presentarse.

#### *Minería de Datos*

Es un mecanismo de explotación que consiste en búsqueda de información valiosa en grandes volúmenes de datos. Trabaja a nivel de conocimiento con el fin de descubrir patrones, relaciones, reglas asociaciones o incluso excepciones útiles para la toma de decisiones. Minería de datos nos ayudaría para descubrir comportamientos de fraude o inusuales para el uso de tarjeta de crédito, por ejemplo.

### Asegurar el existente y nuevo conocimiento

Se refiere a hacer el conocimiento accesible para aquel que lo necesite. Para esto las tecnologías que pueden brindar soporte son:

#### *Bases de datos*

Es una colección de información organizada, las más comunes son las bases de datos relacionales que permiten manejar índices para una mejor y más rápida extracción de la información.

#### *Biblioteca Digital*

Una Biblioteca Digital es un repositorio que guarda el contenido digitalizado en algún formato electrónico de libros y publicaciones de papel, contrarrestando con esto alguna de las desventajas de las bibliotecas tradicionales, donde el papel se degrada y se daña fácilmente, el espacio puede ser una limitante, se requiere personal que admin-



istre, clasifique y ordene los libros manualmente y los usuarios tienen que ajustarse a horarios y disponibilidad de los materiales. La principal ventaja es que una obra en formato electrónico puede ser utilizada por una gran cantidad de personas, además de que existen mecanismos para poder realizar búsquedas sobre los documentos de manera electrónica, de tal forma que vuelve más eficiente la consulta de la información.

#### *Repositorio de documentos*

Pueden ser archivos texto, gráficas, hojas de cálculo, etc. Cualquier documento que tenga información relevante para la organización. Para que sean utilizados deberán ser compartidos en un lugar común con acceso para el resto de la organización.

### **Distribución del conocimiento**

Es compartir el conocimiento, hacerlo llegar a donde pueda ser útil. Las tecnologías que facilitan la distribución son:

#### *Software de trabajo colaborativo*

Incluye sistemas tales como los programas de apoyo a reuniones a distancia, como "chats", groupware, espacios comunes de trabajo, etc.

#### *Servicios Web*

Son un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la Web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer unos servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web. Estos servicios proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. [10] Algunos ejemplos de servicio Web serían: la consulta del inventario o del precio de un artículo en venta, el tipo de cambio para cierta moneda, el clima, la hora, el valor de una acción.

#### *Páginas Web*

Otra fuente de información muy valiosa es todo el universo de información que existe en Internet, en los pocos años que tiene de haberse desarrollado es la biblioteca más grande del mundo, sin duda. Las páginas Web almacenan conocimiento que frecuentemente se está actualizando, existen motores de búsqueda o mecanismos para identificar la adición de nuevo conocimiento en Internet que puede ser explotado.

#### *Redes de servicios de información y conocimiento*

Que incluye servicios de alerta y monitoreo de eventos de interés donde se utilizan

técnicas de inteligencia artificial como son los agentes inteligentes que se describen más adelante.

*Apoyo a la evaluación del capital humano en la toma de decisiones*

Mediante la implantación de sistemas computacionales capaces de manejar y actualizar "mapas" de distribución del conocimiento de la empresa, es decir quiénes saben hacer qué en la empresa.[7]

### **Combinación conocimiento**

Se refiere a aprovechar el conocimiento de múltiples áreas y combinarlo. Las tecnologías que facilitan la distribución son:

*Balanced ScoreCard*

Ayuda a responder preguntas del tipo what-if. En base al conocimiento existente, ayuda a predecir el resultado frente a posibles escenarios. Por ejemplo, como se afecta el costo final de producir determinado producto si se incrementa la materia prima o si sube el costo de la gasolina.

## **2.3. Sistemas Multi-agentes**

Antes de hablar de sistemas multi-agentes, veamos la definición de Agente según Wooldridge, Jennings (1995) es: "Un agente es un sistema computacional que está situado en un ambiente y que es capaz de realizar acciones autónomas en ese ambiente con el fin de lograr el objetivo para el que fue diseñado"

En el contexto de la inteligencia artificial, el término agente inteligente se refiere a una entidad que funciona continuamente y de manera autónoma, en un ambiente en el cual otros procesos toman lugar y en donde existen otros agentes.[10]

Shoham [6] propone que los atributos de un agente son los que se muestran a continuación en el Cuadro ??:

Atributo	Significado
Autónomo	Capacidad de existencia independiente; tiene control de sus propios actos.
Reactivo	Detecta cambios en su entorno y responde inmediatamente.
Orientado a metas	Toma iniciativas para alcanzar sus metas; proactivo.
Persistente	Continuamente corre procesos.
Social	Interactúa con otros agentes y/o usuarios.
Inteligente	Habilidad para aprender y razonar.
Móvil	Habilidad para transportarse a sí mismo de una máquina a otra.

Cuadro 2.1: Atributos de un agente

Los agentes pueden trabajar como entidades “inteligentes” a quienes se delegan ciertas funciones, y que pueden interactuar con otros agentes, cooperando o compitiendo para lograr ciertas metas [3]. En este último caso se habla de sistemas multi-agentes (SMA). Tomando en cuenta que en la tesis proponemos un SMA como soporte a ciertas funciones de la administración de conocimiento, a continuación se presentan algunas de las características de esos sistemas.

Un SMA está generalmente construido de un grupo de agentes autónomos diseñados de forma independiente. Los agentes pueden interactuar debido a un nivel de comunicación de conocimiento entre ellos que les permite completar tareas complejas. Los agentes tienen dos tipos de conocimiento, el social y el individual. El comportamiento social es crucial para que los SMA logren tener inteligencia social. [16]

La comunicación entre agentes se logra por medio del lenguaje estándar ACL (Agent Communication Language) por sus siglas en inglés. ACL brinda una forma eficiente para que los agentes compartan conocimiento, comuniquen el estado mental en que se encuentran y coordinen sus comportamientos. Otro lenguaje estándar que existe pero que ha entrado en desuso es KQML (Knowledge Query and Manipulation Language).

La estructura típica de un SMA es que contiene un número de agentes, que interactúan unos con otros a través de comunicación. Los agentes tienen la capacidad de actuar en el ambiente; diferentes agentes pueden tener diferentes esferas de influencia, en el sentido de que pueden tener control - o influencia - en diferentes partes del ambiente. Estas esferas de influencia pueden coincidir en algunos casos. [12]

Los SMA pueden asistir a las personas en actividades de las redes informáticas de conocimiento. Por ejemplo, agentes que busquen y manipulen datos específicos o coleccionen información de varias redes interconectadas, agentes que clasifiquen el conocimiento para su eficiente localización, etc.

Para llevar a cabo la comunicación entre agentes se han desarrollado lenguajes estándares de comunicación. KQML (Knowledge Query and Manipulation Language) es el primero de ellos y más recientemente ACL (Agent Communication Language). Este último es el utilizado en el presente trabajo de tesis.

### 2.3.1. FIPA-ACL

FIPA-ACL es un lenguaje de comunicación desarrollado por “The Foundation for Intelligent Physical Agents” en 1999 como un estándar para implementar plataformas de SMA basado en el estándar ACL. [12]

Un mensaje FIPA-ACL se vería como sigue:

```
(inform
      :sender agent1
      :receiver agent2
      :content (price auto_escalas 150)
      :language sl
      :ontology subasta_autos
)
```

A continuación se describen las preformativas proporcionadas por el lenguaje de comunicación FIPA-ACL [12]:

**Accept-proposal:** permite a un agente anunciar que acepta la propuesta hecha por otro agente.

**Agree:** se utiliza para que un agente indique que ha aprobado el requerimiento hecho por otro agente. Indica que el que envía el mensaje de “agree” pretende ejecutar la acción solicitada.

**Cancel:** esta preformativa va seguida de un mensaje de “request” previo, indica que ya no se desea la ejecución de una acción en particular.

**Cfp(Call for proposal):** es usada para iniciar la negociación entre agentes.

- Confirm:** permite que el que envía el mensaje pueda validar la veracidad del contenido del mismo.
- Disconfirm:** indica al receptor que el que envía está inseguro del contenido del mensaje.
- Failure:** permite a un agente indicar a otro que el intento de realizar una acción falló (generalmente una acción que fue solicitada previamente).
- Inform:** es el mecanismo básico para comunicar información, el contenido es una declaración. La idea es que el que envía el mensaje quiere que quien lo reciba crea su contenido, implícitamente está indicando que el que envía el mensaje cree en su contenido. Es una de las principales preformativas de FIPA-ACL junto con "request".
- Inform-if:** pregunta implícitamente que una declaración en particular es cierta o falsa.
- Inform-ref:** similar a la anterior, la diferencia es que el agente pregunta en este caso por el valor de una expresión.
- Not-understood:** se utiliza para que un agente le indique a otro que sí reconoce que realizó cierta acción, pero no entiende por qué fue realizada esa acción.
- Propagate:** esta preformativa consta de dos partes: otro mensaje y una expresión que denota un grupo de agentes. La idea es que el receptor del mensaje "propagate" pueda enviar el mensaje embebido a los agentes denotados en la expresión.
- Propose:** permite a un agente hacer una propuesta a otro agente, por ejemplo a un mensaje "prop" que fue enviado previamente.
- Proxy:** permite que el que envía el mensaje utilice al receptor del mismo como un proxy para un grupo de agentes. Consta de dos partes: un mensaje embebido y la especificación de los agentes a los que se desea que se reenvie el mensaje.
- Query-if:** permite que un agente le pregunte a otro si una sentencia es cierta o falsa.
- Query-ref:** es utilizada por un agente para determinar el valor específico para una expresión.
- Refuse:** es usada para que un agente le notifique a otro que no va a realizar cierta acción. El mensaje consta de dos partes: la acción y una sentencia explicando por qué el agente no va a realizar la acción.
- Reject-proposal:** es usada por un agente para indicarle a otro que no acepta una "proposal" que fue hecha como parte del proceso de negociación.

**Request:** otra de las performativas básicas, permite a un agente solicitar a otro que realice alguna acción.

**Request-when:** consta de dos partes: la acción y una declaración, el que envía desea que la acción sea realizada cuando la declaración sea verdadera.

**Request-whenever:** similar a “request-when”, la idea que que el que recibe el mensaje pueda realizar la acción que se le solicita en el momento en que la declaración sea verdadera.

**Subscribe:** el contenido es una sentencia, el sender desea ser notificado cuando suceda algo relacionado con los cambios en la sentencia.

En el Cuadro 2.2 con las performativas de FIPA-ACL clasificándolas por el tipo de función de que se trate [12]:

Performativa	Pasar Información	Solicitar Información	Negociar	Realizar Acciones	Manejar Errores
accept-proposal			x		
agree				x	
cancel		x		x	
cfp			x		
confirm	x				
disconfirm	x				
failure					x
inform	x				
inform-if	x				
inform-ref	x				
not-understood					x
propagate				x	
propose			x		
proxy				x	
query-if	x				
query-ref	x				
refuse				x	
reject-proposal			x		
request				x	
request-when				x	
request-whenever				x	
subscribe		x			

Cuadro 2.2: Performativas de FIPA-ACL clasificados por función

Para hacer que los agentes se entiendan entre sí, no sólo es necesario que hablen el mismo lenguaje sino también que tenga una ontología común. Una ontología es parte

del conocimiento base del agente que describe el tiempo de cosas con las que el agente puede tratar y cómo está relacionado con otros agentes. [19]

### **2.3.2. Ontología**

Una ontología es una especificación de objetos, conceptos y relaciones en un área de interés. Una ontología es más que una taxonomía de clases (o tipos), la ontología también debe describir relaciones. Una ontología se puede análoga al esquema de una base de datos, no al contenido de la base de datos en sí. [20]

En el contexto de agentes, los agentes pueden representar su conocimiento con el vocabulario de una ontología específica. Todos los agentes que comparten la misma ontología tienen un entendimiento de las palabras que utilizan los agentes en su lenguaje de comunicación. Por ejemplo, si tenemos un SMA de subastas de autos, términos como: precio, color, tamaño, barato, caro; son entendidos de la misma forma por todos los agentes que comparten la misma ontología.

### **2.3.3. Herramientas de Implementación de Sistemas Multi-agentes**

A continuación se describen algunas de las principales herramientas de implementación para sistemas multi-agentes basados en el estándar de FIPA-ACL y que son de código abierto:

#### **JADE**

JADE (Java Agent DEvelopment Framework) es una plataforma desarrollada en Java que simplifica la implementación de sistemas multi-agentes a través de un middleware que cumple con las especificaciones de FIPA y cuenta con una serie de herramientas gráficas que facilitan las fases de desarrollo e implementación.[28]

La plataforma multi-agentes puede ser distribuida en distintas máquinas (sin la necesidad de compartir el mismo sistema operativo) y la configuración puede ser controlada remotamente mediante una interfase GUI remota. La configuración puede ser cambiada en tiempo de ejecución moviendo los agentes de una máquina a otra cuando sea requerido.

JADE puede ser considerado un “agent middleware” que implementa una plataforma multi-agentes y un marco de desarrollo. Maneja todos aquellos aspectos que no son peculiares de los agentes que son independientes de las aplicaciones, tales como mensajes, codificación o ciclos de vida de los agentes.

La plataforma puede ser distribuida en múltiples hosts como se muestra en la Figura 2.2. sólo una aplicación java y una JVM es ejecutada en cada host. Cada JVM es básicamente un contenedor de agentes que provee un ambiente completo para la ejecución de agentes y permite que múltiples agentes sean ejecutados concurrentemente en el mismo host.

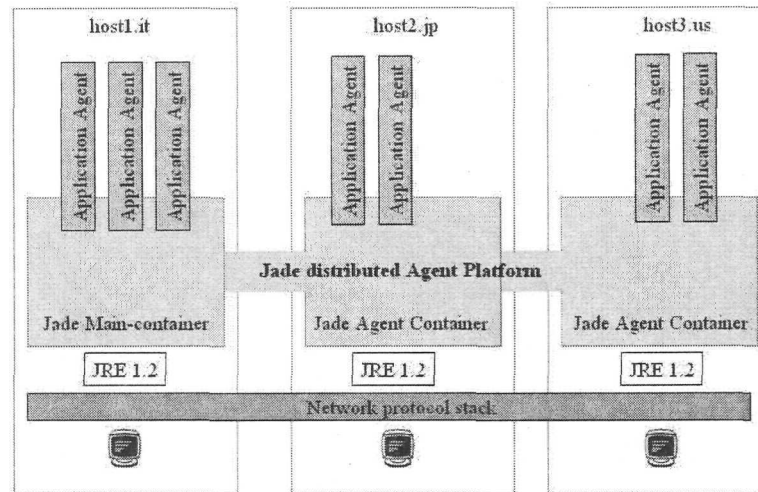


Figura 2.2: Arquitectura distribuida de JADE

La arquitectura de comunicación que se muestra en la Figura 2.3 ofrece mensajería flexible y eficiente, donde JADE crea y administra una cola de mensajes ACL, privada para cada agente. Los agentes pueden acceder las colas combinando varios modos: bloqueo, poleo, timeouts y cumplimiento de patrones. Se ha implementado el modelo completo de comunicación de FIPA.

JADE proporciona algunos servicios de acuerdo a las especificaciones de FIPA, entre las que se encuentran los servicios del Directory Facilitator (DF) y el Agent Management Systems (AMS).

El DF es el responsable de crear una sociedad peer-to-peer y está encargado de administrar el directorio o páginas amarillas. El servicio de páginas amarillas consiste en proporcionar la dirección de un agente dada una o más de sus capacidades. El DF proporciona los identificadores de agentes que corresponden a la descripción de un servicio y a la ontología especificada.

El AMS se encarga de administrar las páginas blancas. El servicio de páginas



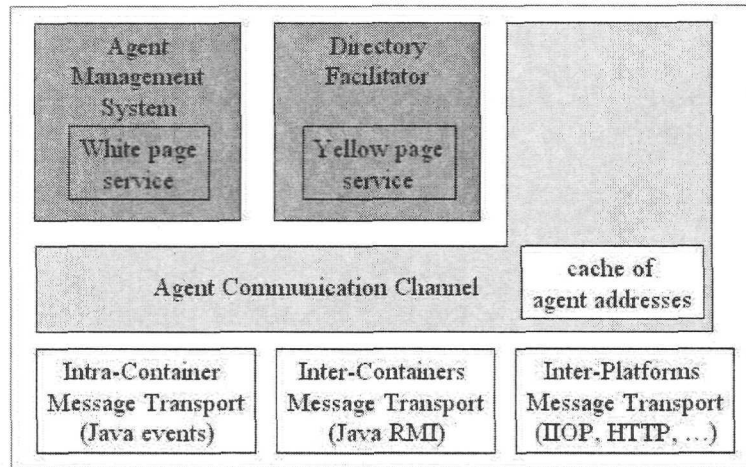


Figura 2.3: Arquitectura de comunicación de JADE

blancas consiste en proporcionar la dirección de un agente dado su nombre. En la Figura 2.4 se muestra la arquitectura interna de JADE:

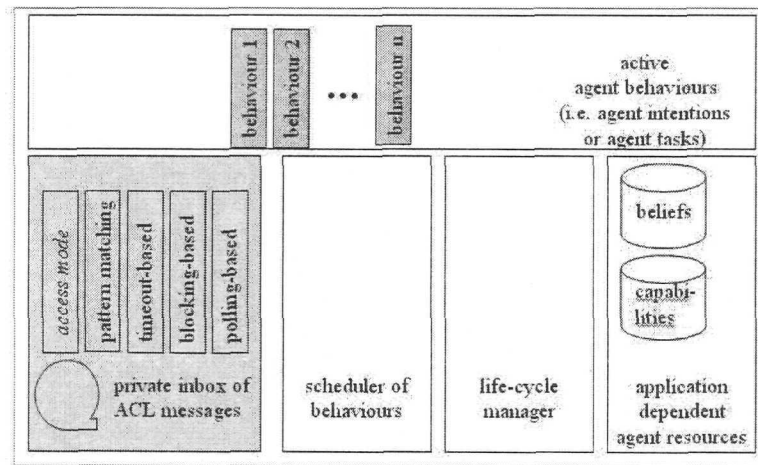


Figura 2.4: Arquitectura interna de JADE

## ZEUS

Zeus proporciona un ambiente de desarrollo gráfico en Java para construir sistemas de agentes distribuidos.[29]

La herramienta de Zeus consiste en tres grupos principales: Agent Component Library, Agent Building Software y Utility Agents. Que se muestran en la Figura 2.5:

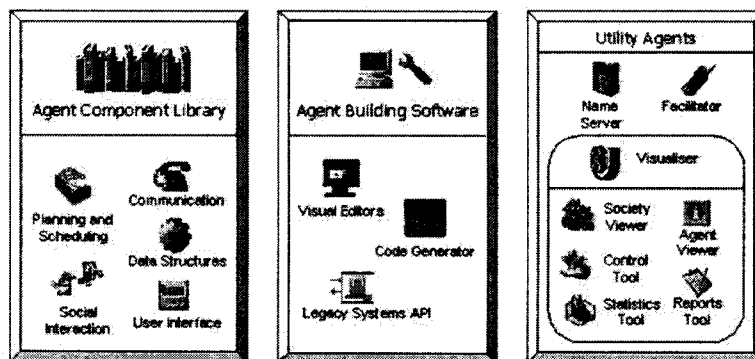


Figura 2.5: Componentes de ZEUS

El Agent Component Library contiene las clases que definen la comunicación, ontología e interacción social entre los agentes. Utiliza KQML como lenguaje de comunicación entre agentes y ACL para la definición de ontologías.

El Agent Building Software, contiene una metodología para guiar al desarrollador en el análisis y diseño del SMA y brinda una interfase gráfica para construirlos.

Utility Agents, son un conjunto de funciones para facilitar la comunicación entre agentes, cuenta con agentes para localizar a otros agentes, visualizar o hacer el rastreo de agentes, por mencionar algunos.

La estructura conceptual de Zeus se presenta a continuación en la Figura 2.6:

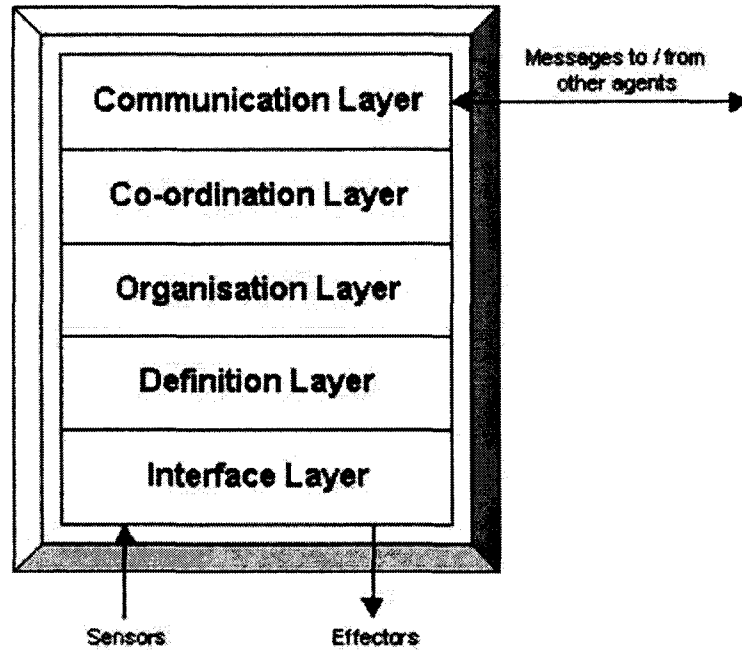


Figura 2.6: Estructura de ZEUS

Mencionándolas de abajo hacia arriba, estas capas son:

**Interface Layer** o Capa de Intefaz, que permite al aente estar ligado con programas externos que lo proven de recursos o desarrollan sus competencias.

**Definition Layer** o Capa de Definición, donde el agente es visto como una entidad autónoma que razona.

**Organisation Llayer** o Capa de Organización, donde el agente es visto en término de sus relación con otros agentes.

**Co-ordination Layer** o Capa de Coordinación, donde el agentes es visto como una entidad social que interactúa de acuerdo al conocimiento de protocolos y estrategias.

**Communication Layer** o Capa de Comunicación, la cual implementa los protocolos y mecanismos que soportan la comunicación entre agentes.

La Capa de Interfaz recibe la entrada de información por medio de sus sensores y tiene la capacidad de cambiar el mundo exterior por medio de sus efectores. Por ejemplo, un sensor puede recibir instrucciones de una interfaz de usuario o bien, el

agente puede actuar modificando una base de datos. Como los sensores y efectores son externos al agente, la configuración de la Capa de Interfaz consiste en especificar las interfaces entre ellos y el agente.

En la Capa de Definición el agente es visto como una entidad autónoma que razona, en términos de sus capacidades de razonamiento, objetivos, recursos, habilidades, creencias y preferencias. Esta capa es físicamente ejecutada por las clases del Agent Component Library de ZEUS. En esta etapa de la metodología se configuran dichos componentes.

En la Capa de Organización el agente es visto en términos de su relación con otros agentes. Esto introduce el concepto de una Agencia (un grupo de agentes relacionados). Las agencias pueden ser reales, como sería en el caso de compartir un atributo virtual en común, por ejemplo ser parte de la misma compañía. O las agencias pueden ser virtuales, en este caso los que la componen tienen un acuerdo de cooperación. En este paso de la metodología se configuran los agentes en términos de las agencias a las que pertenecen, el rol que juegan en la agencia, que otros agentes están enterados de ese rol y que habilidades conoce sobre otros agentes.

En la Capa de Coordinación el agente es considerado una entidad social, por lo tanto etapa involucra configurar al agente de acuerdo al comportamiento social deseado. Esto se logra equipando al agente con protocolos de negociación y estrategias.

La Capa de Comunicación proporciona un protocolo de transporte y un lenguaje que permite que los agentes puedan enviarse mensajes entre sí. Por compatibilidad esta capa debe ser la misma para cada agente, no es necesario que el desarrollador configure esta capa.

El proceso de configuración se hace utilizando la herramienta de ZEUS Agent Generator, la cual generará el código fuente del agente. El desarrollador sólo implementa la funcionalidad específica a la aplicación.

## **2.4. Proyecto RICA (antes CORREA)**

El proyecto RICA (Redes Informáticas de Conocimiento con Agentes, antes llamado CORREA) propone un conjunto integrado de herramientas de apoyo a la Administración del Conocimiento, usando como base la tecnología de Agentes Inteligentes.[14]

Para las pruebas experimentales de RICA se seleccionó el entorno de investigadores

en computación en México. En este entorno RICA busca desarrollar herramientas de administración de conocimiento que brinden apoyo a la interacción e integración de los grupos de investigadores en computación en el país.

Los servicios propuestos en RICA son:

- Definición de una taxonomía de áreas de interés para un grupo de usuarios.
- Notificación automática, por medio de correo electrónico o de un sistema “visualizador”, de modificaciones a páginas Web de interés para los usuarios.
- Interrogación elemental de una base de datos distribuída de investigadores del algún área determinada.
- Repositorio digital distribuido de reportes de investigación.
- Soporte a avisos indexados por interés de investigación, de manera que esos avisos lleguen a los investigadores de un área determinada.

La arquitectura de RICA comprende los siguientes elementos:

1. Los usuarios se conectarán por medio de máquinas “cliente” (del tipo PC) a la máquina “servidora” de poder mediano (servidor Unix por ejemplo) que nunca se apague, pues es necesario mantener corriendo una serie de procesos que son el corazón de los servicios de RICA. Llamamos a estos servidores “sites”.
2. RICA está constituido por una serie de “sites” interconectados, que son agentes inteligentes que intercambian mensajes, y que se coordinan para ofrecer una forma transparente a sus usuarios una red de servicios de apoyo. Esta estructura se muestra en la Figura 2.7:

Las computadoras de los sites deben estar prendidas permanentemente y corren procesos que operan 24 horas. Cuando un usuario se conecta al sistema mediante su PC recibirá informaciones colectadas por el “site” en su ausencia.

Una parte del software de RICA corre en los “sites” (la mayor parte), mientras que la interfaz de usuario corre en una máquina de tipo personal.

Los componentes de software que integran a cada “site” son:

- Un motor de inferencia, que permite realizar deducciones a partir de los hechos registrados en su base de hechos.

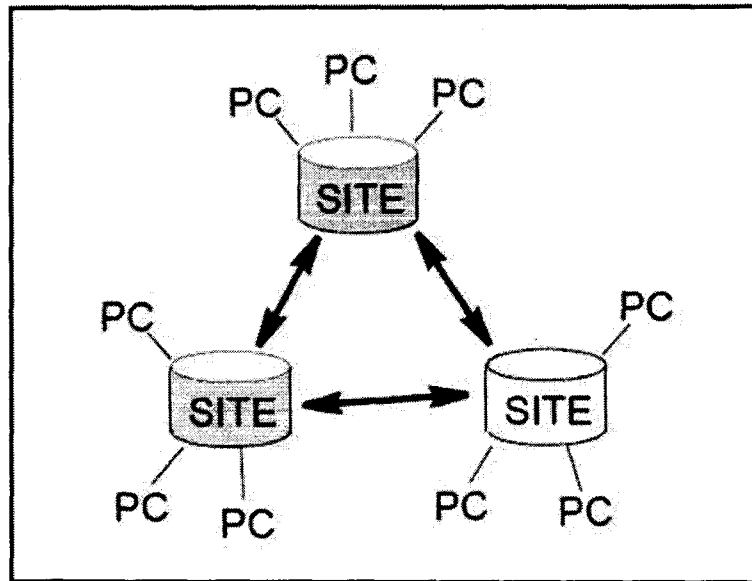


Figura 2.7: Arquitectura de RICA

- Una base de datos, que almacena en forma permanente las informaciones de los usuarios, de sus áreas de interés, de los servicios solicitados, etc. La base de datos guarda toda la información de RICA de modo que es posible reestablecer los servicios en caso de una caída del sistema.
- Un administrador de objetos, que carga informaciones de la base de datos y las mantiene disponibles para ser consultadas por otros módulos de RICA.
- “Adaptadores”, que realizan alguna función de acceso al exterior de RICA para implementar algunos servicios. Un “actualizador”, que mantiene la coherencia entre las informaciones de la base de datos y las que están en el motor de inferencia y en los objetos compartidos.

La relación entre estos componentes se ilustra en la Figura 2.8:

Como se aprecia en el diagrama, un “browser” de internet en la máquina del cliente despliega la interfaz de usuario y se comunica con un programa tipo “script” que corre en el servidor Web del site (se usa un paquete llamado W3, parte de mSQL que tiene un script genérico y un lenguaje de comandos propio).

El programa tipo “script” se comunica a su vez con la base de datos con la cual mantiene una comunicación a nivel socket donde ejecuta al “actualizador” el cual incorpora las modificaciones a la base de datos tanto en el Motor de Inferencia como en el Administrador de Objetos. Durante la operación normal del sistema, el Motor de

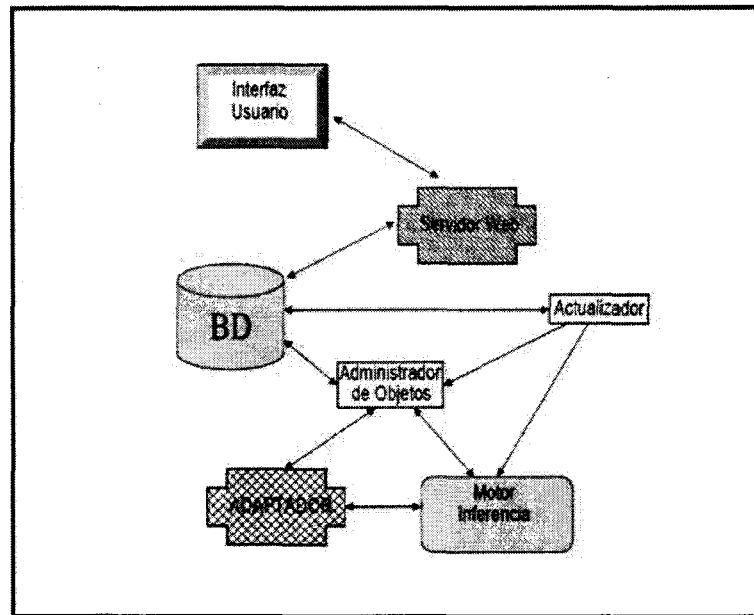


Figura 2.8: Componentes de de RICA

Inferencia es quien lleva la lógica de implementación de los servicios de RICA.

La base de datos relacional utilizada en RICA es miniSQL y la herramienta utilizada para diseñar el motor de inferencia es JESS (Java Expert System Shell).

## Capítulo 3

### Arquitectura multi-agentes para JITIK

En este capítulo se presenta una arquitectura multi-agente para soportar los servicios de distribución de la información y conocimiento entre varias entidades.

La información y el conocimiento pueden estar contenidos en varias fuentes; con la gran aceptación que tienen los sistemas informáticos en la actualidad es común encontrar que el conocimiento y la información están resguardados en medios electrónicos. Sin embargo, la gran cantidad tanto de uno como de otro y los múltiples repositorios donde se almacena hace complejo el proceso de distribución.

En la Figura 3.1 se muestra conceptualmente como una arquitectura de SMA puede brindar el soporte para esos servicios y así permitir el intercambio oportuno de información y conocimiento de distintas fuentes y hacia distintos usuarios.

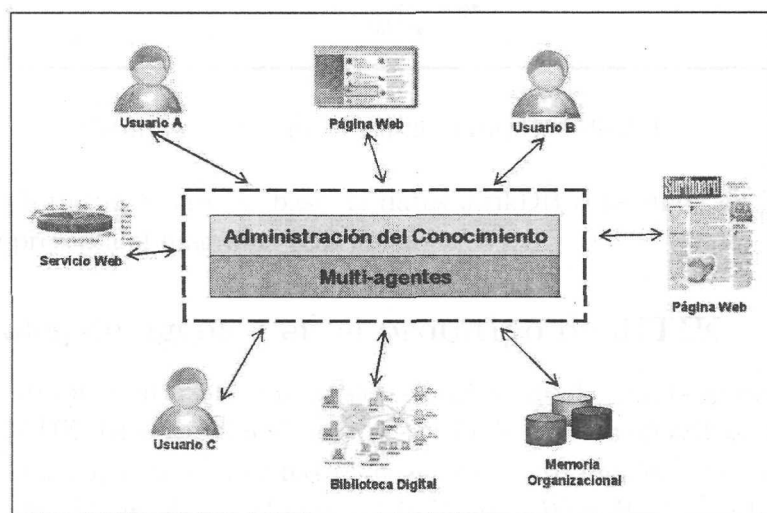


Figura 3.1: Arquitectura conceptual de JITIK

La figura muestra algunos repositorios donde puede encontrarse la información y el conocimiento, como son páginas Web, servicios Web, bibliotecas digitales y memorias



organizacionales, también muestra algunos usuarios interesados en tener acceso a esos repositorios. Una arquitectura de sistema multi-agentes puede ser la que brinde la distribución de esas piezas de conocimiento e información y hacerlas disponibles para los usuarios.

### 3.1. Descripción General de Sistema JITIK

JITIK significa “Just In Time” Information and Knowledge y tiene como objetivo principal asegurarse que la información relevante para una organización esté al alcance de las personas adecuadas en el momento preciso.

Un aspecto a considerar es la categorización de personas, conocimiento e información la cual se hace a través de la combinación de varias jerarquías de conceptos.

Por ejemplo, una persona particular en una organización puede ser categorizada por:

- La división o área en el departamento, tomado de una clasificación por áreas (esto puede seguir o al menos aproximar el organigrama de la empresa).
- El nivel de responsabilidad de la persona, típicamente tomada del organigrama de la empresa.
- Algunos intereses y preferencias adicionales del individuo.

El conocimiento también debe ser clasificado en base a una jerarquización y a los distintos repositorios donde se encuentre. Por ejemplo, en el ambiente académico los profesores pueden ser situados en la estructura lógica de la universidad, o sus departamentos, los cuales a su vez son categorizados por medio de un grupo de áreas de conocimiento o disciplinas, tales como “Ingeniería Mecánica”, “Ciencias Computacionales”, etc.

Adicionalmente, los profesores tienen intereses académicos, los cuales pueden ser categorizados por algunos estándares de clasificación, tal como se utiliza en las bibliotecas. Los estudiantes pueden ser categorizados primero por el grado que cursan o bien por los cursos específicos que están tomando. El personal administrativo de la universidad puede ser categorizado por el tipo de trabajo que realizan, por los clientes que tienen, por el nivel en la jerarquía, etc.

Otro aspecto a considerar es el “momento preciso”, el cual es tomado en cuenta por JITIK como la ocurrencia de un “evento”. Los eventos en JITIK pueden ser generados de distintas maneras:

- Un usuario genera intencionalmente un evento - cuando ella/él quiere, por ejemplo, que un comunicado escrito sea distribuido por el sistema -.
- Algunos procesos internos en la organización pueden generar eventos. Por ejemplo, en la biblioteca de la universidad, cuando se agrega un libro al catálogo, los profesores interesados en el área relacionada con el libro deben ser notificados. Por supuesto, estamos asumiendo que los procesos internos son computarizados o que permiten alguna manera de rastrear la información en el ambiente de sistemas de la organización, así puede ser generado el evento correspondiente en JITIK.
- Los procesos externos de la organización son monitoreados periódicamente en la búsqueda de alguna condición específica, como es la modificación de una página Web, el valor de una variable en la base de datos, etc.
- El simple paso del tiempo puede ser considerado, en algunos casos, como un evento. Por ejemplo, para algunos ajustes se requiere una respuesta después de algún periodo, y si el sistema rara vez falla para completar la respuesta, el sistema debe enviar al usuario un recordatorio con un extracto del ajuste correspondiente.

En conclusión, de esta manera JITIK es un método para relacionar una pieza de conocimiento o información, especificado como se mencionó anteriormente, a un grupo de usuarios cuando se produzca el evento.

Este sistema podría ser utilizado en distintos escenarios, ya sea en el ambiente empresarial, académico o incluso para entretenimiento. Enseguida se plantean algunos ejemplos de escenarios donde aplicaría.

#### **Escenario A. Aviso de noticias relacionadas con la competencia**

**CONTEXTO:** Una empresa de tecnología que vende un producto de seguridad los vendedores tienen que estar informados de los nuevos productos o mejoras que realiza la competencia para lograr posicionar mejor sus productos.

**SITUACION ACTUAL:** Los vendedores se enteran generalmente de lo que está haciendo su competencia por que algún cliente les pregunta sobre cierta funcionalidad o producto que les ofrecieron. Entonces el vendedor acude a investigar qué es lo que está haciendo su competencia y si sus propios productos le pueden ofrecer una alternativa. Pero la respuesta puede tomar tiempo y mientras tanto eso le quita oportunidad de competir.

**USO DE SMA:** El sistema podría permitirle al vendedor suscribirse a servicios de monitorio de páginas Web, de noticias, revistas informáticas o bien, en la memoria or-

ganizacional donde aparezca el nombre de su competencia y así recibir una notificación con las noticias que se están generando, así como de comparativos. De tal forma que le permite estar enterado de los movimiento de su competencia y poder anticipar los requerimientos de sus clientes.

### **Escenario B. Coleccionista de modelos de autos a escala**

**CONTEXTO:** Una persona interesada en coleccionar autos modelos a escala desea saber oportunamente del lanzamiento, la venta o subasta de los autos más codiciados.

**SITUACION ACTUAL:** El coleccionista de autos a escala tiene que buscar en revistas, tiendas, páginas de Internet o de subastas para enterarse si está disponible el modelo que le interesa.

**USO DE SMA:** El sistema conocería las características del auto en escala en que está interesada la persona y se encargaría de recolectar información en sitios de subastas, páginas Web y tiendas para poder informarle oportunamente cuando el auto en cuestión esté disponible para venta.

### **Escenario C Nuevo libros en biblioteca**

**CONTEXTO:** La biblioteca registra nuevos libros en su acervo. Es conveniente que los usuarios, en particular los profesores, se enteren de la existencia de las nuevas adquisiciones para que éstas sean mejor aprovechadas.

**SITUACION ACTUAL:** Ocasionalmente las bibliotecas envían a profesores correos electrónicos con listas de libros nuevos, o bien una liga a una lista equivalente. Esto es mucho mejor que nada, pero las listas de libros, aún ordenadas por su clasificación temática, son excesivamente largas, lo que desanima a muchos profesores a revisar dicha lista.

**USO DE SMA:** El sistema permitiría enviar noticias individualizadas a los profesores sobre el incremento en el acervo bibliográfico, de acuerdo con el perfil de intereses del profesor. Dicho perfil considera tanto aspectos institucionales, tales como las materias en que imparte clases o el departamento al que pertenece, como intereses individuales que el mismo profesor ha declarado en el sistema. De esta manera, los profesores son notificados de la existencia de libros que efectivamente corresponden a sus intereses, sin tener que examinar largas listas llenas de informaciones que no le conciernen.

## 3.2. Arquitectura multi-agente de JITIK

Recordemos dos acepciones del término agente:

- El agente visto como una entidad “inteligente” a quien se delegan ciertas funciones.
- Los sistemas “multiagentes”, que trabajan mediante la interacción con otros agentes, cooperando o compitiendo para lograr ciertas metas.

En JITIK se consideran ambos puntos de vista. El usuario delega ciertas tareas a agentes de JITIK, como el monitoreo de páginas Web, el envío de mensajes, etc., en lugar de hacerlo él mismo. Igualmente están contempladas actividades de cooperación entre agentes distribuidos.

Tecnológicamente JITIK está basada en una arquitectura multi-agentes en un ambiente distribuido tal como se muestra en la Figura 3.2:

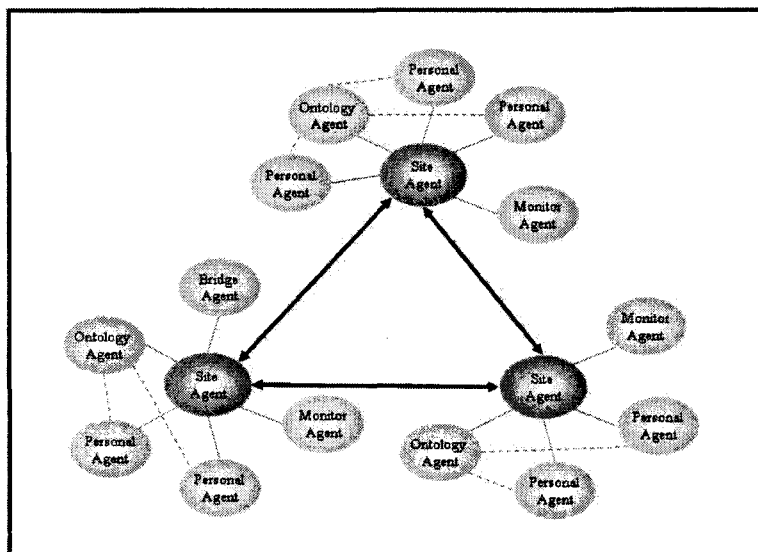


Figura 3.2: Arquitectura distribuida de JITIK

En la figura se presentan tres sitios de JITIK donde la comunicación entre sitios se da a través del Site Agent, que es el componente central de la arquitectura.

Cada sitio tiene un Ontology Agent que es el que tiene la representación del mundo real que comparten todos los agentes del sitio. El Ontology Agent brinda servicio a otros agentes locales.

Cada sitio puede tener ninguno o varios Monitor Agent y Bridge Agent. El Monitor Agent tiene la función de monitorear en tiempo real alguna fuente de información externa de forma proactiva, mientras que el Bridge Agent está encargado de la integración con aplicaciones heredadas o externas sobre las que se pueda tener cierto nivel de control.

Además JITIK aplica mecanismos de inferencia para realizar el proceso de la distribución de información y conocimiento tomando en cuenta los perfiles o intereses de los usuarios y también la relación o jerarquización de áreas de conocimiento.

La que se refiere a notificar un evento que es de interés para los usuarios de JITIK sería aplicada por ejemplo a dar el aviso sobre un nuevo libro del área de Algoritmos Genéticos en la biblioteca de una universidad. Este evento podría ser de interés para maestros y alumnos de las áreas informáticas. Mediante una inferencia JITIK determinaría las personas a las que hay que notificar esa información depende de los intereses predefinidos, la inferencia se representa como sigue:

$$Usuario(U) \wedge InteresadoEn(U, X) \wedge EventoRelacionado(E, X) \Rightarrow Notificar(U, E)$$

Siguiendo con el ejemplo anterior, un usuario tal vez registre su interés en todos los temas relacionados con Inteligencia Artificial. Otra regla de inferencia ayudaría a determinar que si el evento está relacionado al área de Algoritmos Genéticos que pertenece al área de Inteligencia Artificial, entonces el evento también está relacionado con Inteligencia Artificial, para lo anterior se utilizaría la siguiente regla:

$$EventoRelacionado(E, X) \wedge PerteneceA(X, Y) \Rightarrow EventoRelacionado(E, Y)$$

### 3.3. Roles de Agentes

A continuación se define los roles de los agentes que conforman la arquitectura de JITIK en un sitio como se muestra en la Figura 3.3.

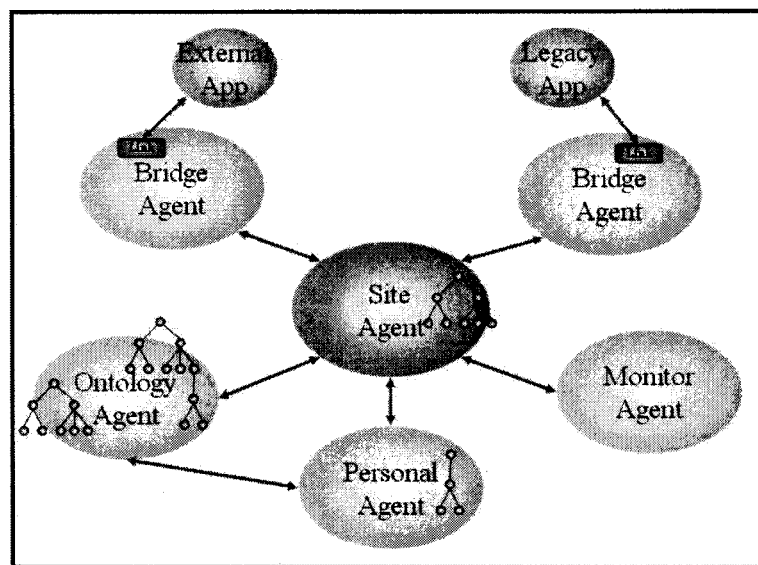


Figura 3.3: Arquitectura de JITIK

#### Site Agent

Es el componente central en la arquitectura de un sitio de JITIK, es el intermediario en algunos de los servicios para los agentes locales y es el que se encarga de comunicarse con los Site Agents de otros sitios de JITIK para que pueda colaborar entre ellos.

Este agente se ayuda en algunos casos de los mecanismos de inferencia del Ontology Agent para determinar a quien debe dirigir los mensajes. En el Cuadro 3.1 se presenta el rol del agente.

Rol	Site Agent, brinda la interacción entre los distintos agentes del sistema
Responsabilidades	Se encarga del intercambio de mensajes entre agentes
Colaboradores	Site Agent, Ontology Agent, Persona Agent, Páginas Amarillas, Páginas Blancas
Interfases externas	-
Relaciones	-
Experiencia	Resolución de problemas, administración de la interacción entre agentes
Interacciones	FIPA-ACL
Otros	Corre de forma persistente, es uno por sitio

Cuadro 3.1: Rol del agente de sitio

### Ontology Agent

El rol de este agente es representar de manera conceptual distintas formas de conocimiento así como su jerarquía y relaciones. Todo aquello del mundo real que quiera ser conceptualizado.

Una parte de los mecanismos de inferencia de JITIK se utilizan en el agente de ontología tal y como se explicará más adelante. En el Cuadro 3.2 se presenta el rol del agente.

Rol	Ontology Agent, administra la ontología común para todos los agentes
Responsabilidades	Responde a consultas sobre la ontología
Colaboradores	Site Agent, Personal Agent, Páginas Blancas
Interfases externas	Base de datos
Relaciones	-
Experiencia	Ontología, consultas
Interacciones	FIPA-ACL
Otros	Corre de forma persistente, es uno por sitio

Cuadro 3.2: Rol del agente de ontología

### Personal Agent

Es el agente que interactúa con el usuario, el usuario puede definir sus intereses, suscribirse a servicios, definir su perfil y también recibir retroalimentación de acuerdo a sus intereses previamente especificados.

Este agente puede filtrar algunos de los mensajes que le mostrará al usuario ayudándose de los mecanismos de inferencia contenido en el Ontology Agent.

Las formas de interactuar con el usuario pueden ser variadas, desde un mensaje a un teléfono celular, un correo electrónico o bien un mensaje mediante una interfaz gráfica del sistema. En el Cuadro 3.3 se presenta el rol del agente.

Rol	Personal Agent, brinda la interacción con los usuarios
Responsabilidades	Maneja la interfaz entre los usuarios y el resto de los agentes en el sistema
Colaboradores	Site Agent, Ontology Agent, Páginas blancas
Interfases externas	Interfaz gráfica, base de datos
Relaciones	-
Experiencia	Administración de la interfase del usuario
Interacciones	FIPA-ACL
Otros	El usuario decide cuando activarlo, es uno por usuario de la arquitectura y está asociado a un site

Cuadro 3.3: Rol del agente personal

### Monitor Agent

Existen otras fuentes de información externa a la organización que son relevantes, tales como sitios Web, repositorios de datos en internet y otros sistemas de monitoreo, sobre las que generalmente no se tiene control.

El objetivo de este agente es recolectar toda la información externa relevante para el objetivo del sistema y el interés de los usuarios en tiempo real y así notificarlo.

Una característica importante de los servicios de monitoreo es que el agente proactivamente es el que toma la iniciativa para revisar la fuente de información externa. Está orientado a tareas que pueden tener un comportamiento periódico. En el Cuadro 3.4 se presenta el rol del agente.



Rol	Monitor Agent, notifica de cambios en fuentes de información externa
Responsabilidades	Monitorear fuentes de información externa y notificarlas a la arquitectura
Colaboradores	Site Agent. Páginas Blancas
Interfases externas	Fuentes externas, base de datos
Relaciones	-
Experiencia	Resolución de problemas
Interacciones	FIPA-ACL
Otros	corre de forma persistente, es uno por cada fuente de información a monitorear

Cuadro 3.4: Rol del agente monitor

### Bridge Agent

Existen otras fuentes de información externa que pueden ser integradas en el sistema. Sobre estas fuentes hay cierto grado de control y se puede buscar la integración a JITIK.

Los servicios de búsqueda en bibliotecas digitales o en memorias organizacionales son el tipo de aplicaciones externas al que un Bridge Agent brindaría acceso.

En este caso la consulta a la fuente de información lo dispara un evento de la aplicación o la petición de consulta por parte de un usuario del sistema. En el Cuadro 3.5 se presenta el rol del agente.

Rol	Bridge Agent, hace consultas en fuentes de información externa
Responsabilidades	Realizar consultas a información externa y notificarlas a la arquitectura
Colaboradores	Site Agent. Páginas Blancas
Interfases externas	Fuentes externas, base de datos
Relaciones	-
Experiencia	Resolución de problemas
Interacciones	FIPA-ACL
Otros	Corre de forma persistente, es uno por cada fuente de información a consultar

Cuadro 3.5: Rol del agente bridge o puente

### 3.4. Servicios de JITIK

Los servicios de una arquitectura multi-agentes, como la que se propone en JITIK, pueden ser muy variados ya que lo que se busca es acercar una pieza de conocimiento a otros agentes o usuarios interesados.

#### Servicio de Páginas Blancas

Es parte del estándar de FIPA- ACL. El servicio de páginas blancas consiste en poder localizar a otro agente dentro de la arquitectura. Todos los agentes hacen uso del servicio de páginas blancas en la arquitectura de JITIK.

En la Figura 3.4 se presenta el diagrama UML (Unified Modeling Language) para los servicios de páginas blancas:

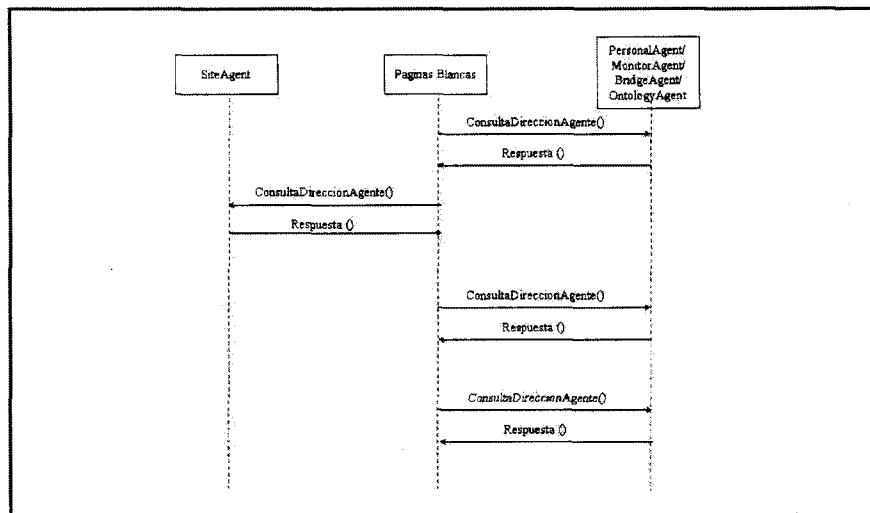


Figura 3.4: Diagrama UML para el servicio de páginas blancas

## Servicio de Páginas Amarillas

Es parte del estándar de FIPA- ACL. Es un servicio donde los agentes se pueden registrar a sí mismos así como sus habilidades y permite que otros agentes hagan búsquedas sobre descripción de servicios para encontrar colaboradores.

El Site Agent es el único que hace consultas sobre las habilidades que tienen los agentes registrados en la plataforma de JITIK.

En la Figura 3.5 se presenta el diagrama UML para los servicios de páginas amarillas:

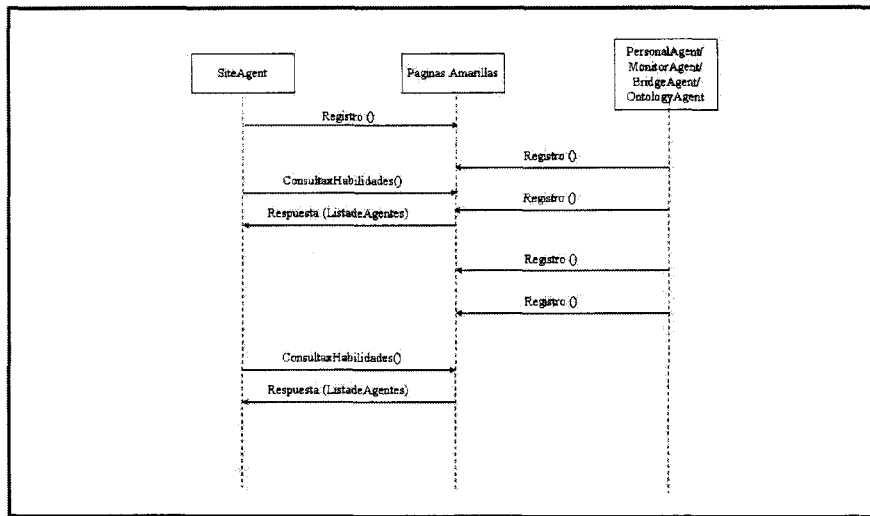


Figura 3.5: Diagrama UML para el servicio de páginas amarillas

**NOTA IMPORTANTE:** Normalmente estos servicios son proporcionados por las herramientas de desarrollo multi-agentes. Por cuestión de simplicidad las consultas a páginas blancas y amarillas quedarán implícitas para el resto de los servicios de la arquitectura.

## Servicio de Ontología

Ontología es la especificación explícita de una conceptualización. Conceptualización es una visión abstracta y simplificada del mundo que queremos representar para cierto propósito [13].

El Servicio de Ontologías hacer consultas sobre la representación del mundo real que es común para los agentes dentro de JITIK.

En la siguiente Figura 3.6 presenta el diagrama UML para el servicio de ontología, donde se muestra que las consultas a la ontología las realizan el Personal Agent y el Site Agent.

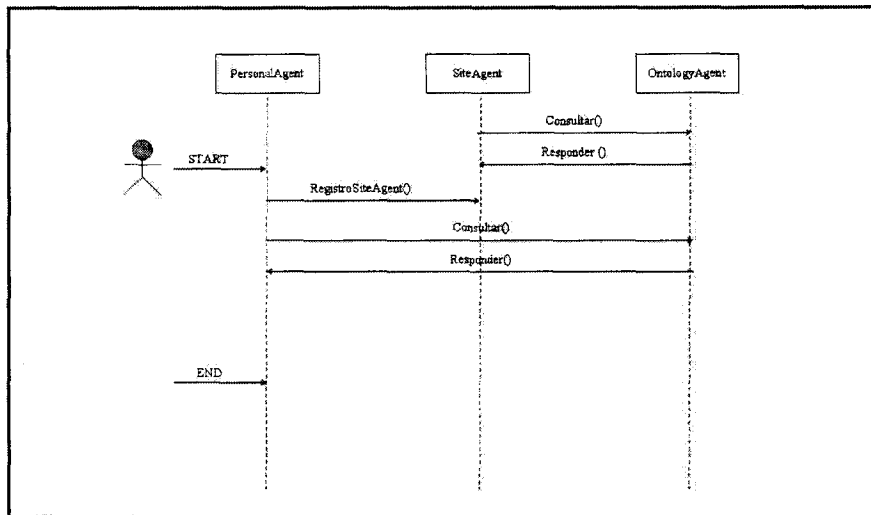


Figura 3.6: Diagrama UML para el servicio de ontología

## Servicios de Mensajería

Permite la interacción entre los distintos servicios y usuarios del sistema. Este servicio puede ser de uno a uno o de uno a muchos usuarios del sistema, dependiendo de los criterios en base a los cuales se haga la distribución.

En la Figura 3.7 se muestra el diagrama UML para la administración de mensajes dentro de la arquitectura, donde se muestra que la entrega de mensajes hacia los usuarios a través del PersonalAgent se realiza a través del SiteAgent. En caso de que el PersonalAgent no se encuentre activo en ese momento se tiene un pizarrón de mensajes para su entrega posterior.

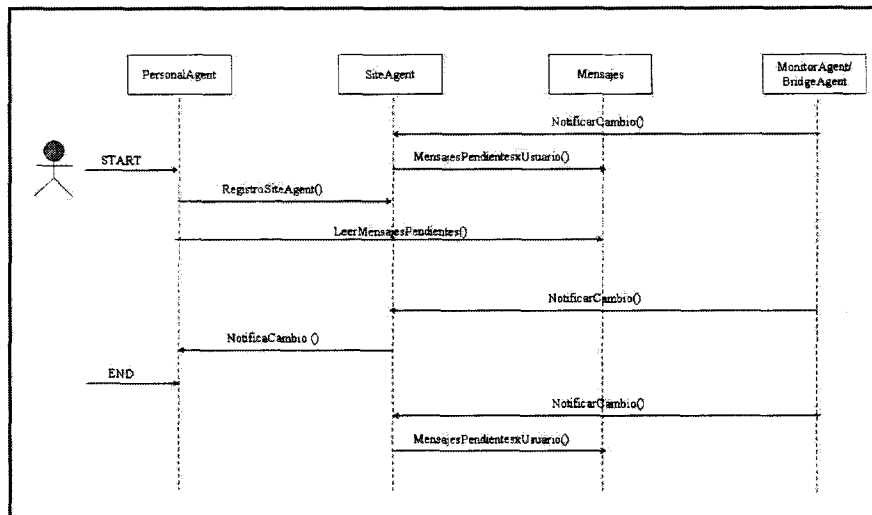


Figura 3.7: Diagrama UML para el servicio de mensajería

## Servicio de Monitoreo

El objetivo de los servicios de monitoreo es estar conectado en tiempo real con recursos disponibles en fuentes externas, como por ejemplo internet. Dentro de esta categoría se distinguen servicios como el monitoreo de páginas Web.

El diagrama UML para el servicio de monitoreo sería el que se muestra en la Figura 3.8, donde el Monitor Agent se encarga de notificar al Site Agent sobre cambios encontrados en el recurso que está monitoreando en el momento en que es detectado. Los usuarios se registran al Site Agent y reciben la notificación de que algún recurso de su interés ha cambiado.

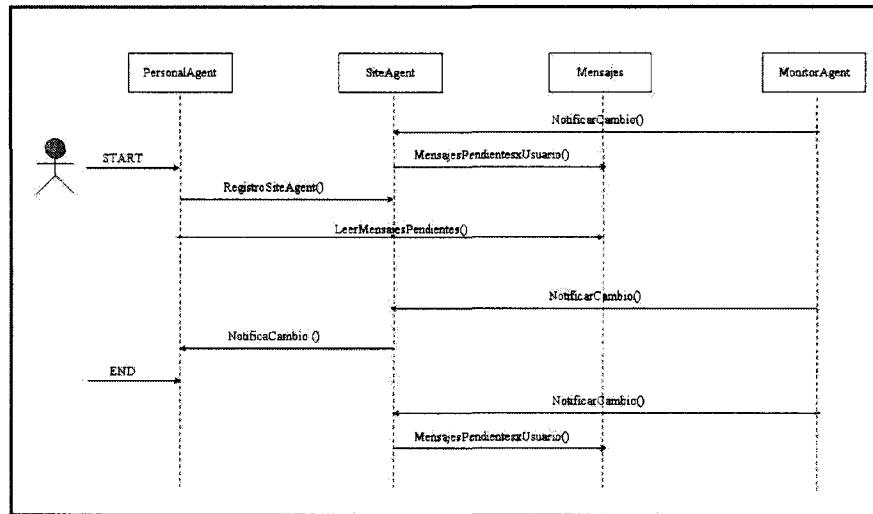


Figura 3.8: Diagrama UML para el servicio de monitoreo

## Servicio de Bridge

El objetivo de los servicios de puente o bridge es lograr la comunicación con aplicaciones existentes o bien externas que sean de interés para el sistema. Hacia estas aplicaciones se pueden disparar consultas sobre la información relevante para el sistema. Dentro de esta categoría se encuentran servicios como la búsqueda de información en bibliotecas digitales.

En la Figura 3.9 se presenta el diagrama UML para el Servicio de Bridge, donde el Usuario dispara la consulta a través del Personal Agent, la solicitud la recibe el Site Agent quien a su vez le hace la petición al Bridge Agent quien se encarga de resolver la petición y regresar el resultado al Site Agent. El Site Agent le notifica la respuesta al Personal Agent.

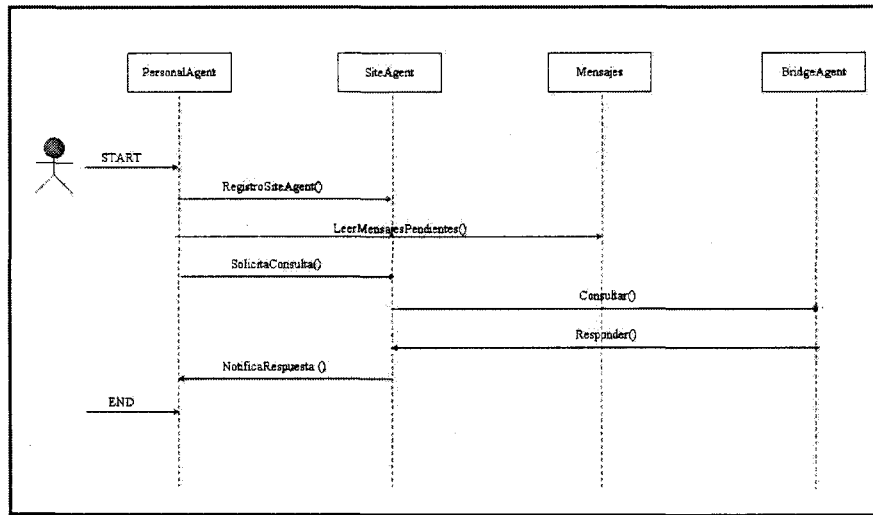


Figura 3.9: Diagrama UML para el servicio de bridge o puente

## Capítulo 4

### Prototipo y pruebas

Los criterios utilizados para seleccionar una herramienta de desarrollo para construir el prototipo de JITIK fueron:

- Que la herramienta fuera de software libre.
- Que se apegara al estándar de FIPA-ACL que es el estándar más actual para lenguajes entre agentes.
- De preferencia que la herramienta tenga servicios básicos de FIPA-ACL ya implementados.
- Que la herramienta muestre potencial de crecimiento y desarrollo.
- Que haya una buena cantidad de usuarios utilizándola.
- Que esté bien documentada y haya forma de obtener soporte en caso de requerirlo.
- Que sea una herramienta sencilla de utilizar.

De las herramientas analizadas como opción para construir el prototipo de JITIK es JADE la que más se acerca a los criterios deseados por las siguientes razones:

- Es una herramienta de software libre.
- Se apega fielmente al estándar de FIPA-ACL.
- Tiene incluidos los servicios de páginas blancas y amarillas.
- Existe una comunidad importante de desarrolladores en JADE que utilizan el código y están aportando mejoras a la herramienta.
- Existe documentación detallada de uso y configuración con ejemplos disponibles.
- Hay un foro de discusión donde están incluidos los creadores de la herramienta que permiten que cualquier duda o problema sea atendido.
- JADE maneja una interfaz gráfica sencilla de utilizar, además tiene un agente Sniffer que permite hacer el rastreo del sistema durante la etapa de desarrollo.



## 4.1. Prototipo de JITIK

Para la implementación del prototipo de JITIK se redujo la arquitectura a un sitio con cuatro agentes y a la implementación de algunos servicios que se explicarán más adelante en esta sección. En el Apéndice A se anexa el pseudocódigo de la implementación del prototipo. En la Figura 4.1 se observa la arquitectura del prototipo con los agentes que lo componen:

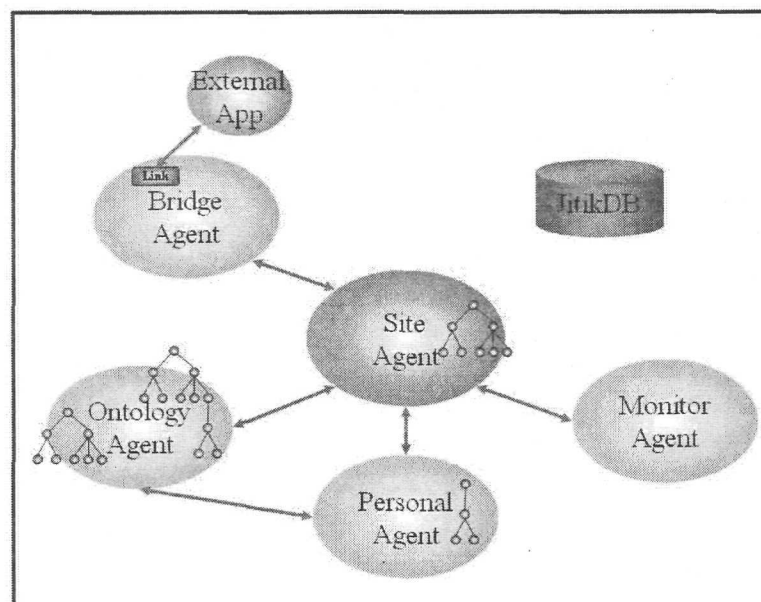


Figura 4.1: Arquitectura del prototipo de JITIK

Para el prototipo se creó la base de datos JitikDB construida en MySQL, en la Figura 4.2 se presenta el diagrama relacional.

### 4.1.1. Roles de agentes en el prototipo de JITIK

En el prototipo se utilizan en total siete agentes, dos de ellos proporcionados por la herramienta JADE: Directory Facilitator (servicio de páginas amarillas) y Agent Management Systems (servicio de páginas blancas) y el resto implementados para llevar a cabo los servicios de Ontología, Monitoreo de Páginas Web, Bridge de Integración con un sistema de Biblioteca Digital y Envío de Mensajes entre usuarios de acuerdo a varios criterios.

Estos seis agentes conforman un sitio, sin embargo pueden ejecutarse más sitios del mismo tipo y hacer que los agentes y usuarios interactúen entre sitios.

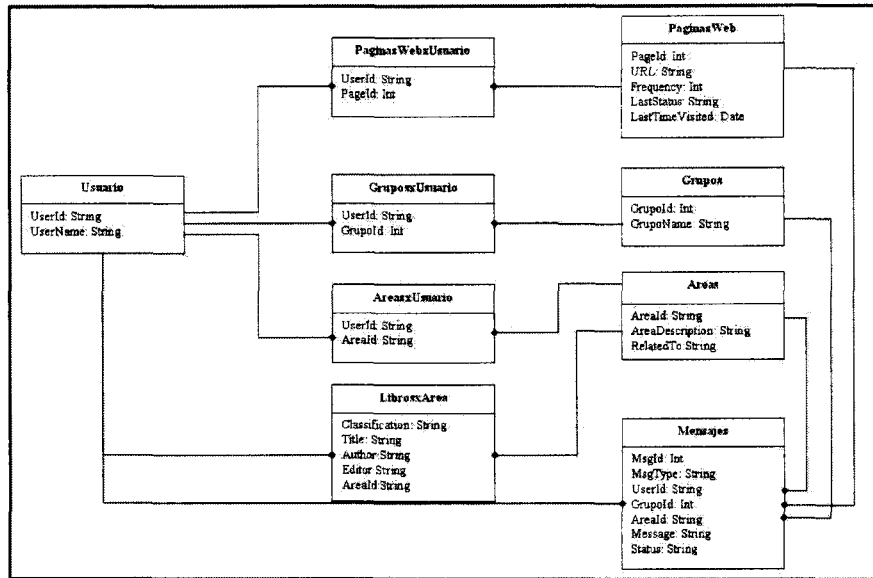


Figura 4.2: Diagrama de base de datos para el prototipo de JITIK

La definición de los roles de los agentes dentro del prototipo son instancias de la definición general que se hizo en la sección anterior. A continuación se describen los mismos en el contexto del prototipo.

### Agent Management Systems

El Agent Management Systems (AMS) también es un servicio brindado por la herramienta JADE y existe uno por sitio. Su rol consiste en proporcionar la dirección de un agente dado su nombre.

Cuando un agente quiere enviarle un mensaje a otro primero consulta con AMS dónde puede localizarlo. En el Cuadro 4.1 se describe el rol del agente.

Rol	Agent Management Systems, responde consultas de páginas blancas del sistema
Responsabilidades	Proporciona la dirección de localización para un agente dado su nombre
Colaboradores	Site Agent, Ontology Agent, Persona Agent, Monitor Agent, Bridge Agent
Interfases externas	-
Relaciones	-
Experiencia	Servicios de FIPA
Interacciones	FIPA-ACL
Otros	Proporcionados por JADE

Cuadro 4.1: Rol del agente Agent Management Systems de JADE

### Directory Facilitator

El Directory Agent (DF) lo proporciona la herramienta JADE y es el encargado de brindar el servicio de Páginas Amarillas al resto de los agentes. Existe uno por sitio. [15]

Básicamente los agentes registran su existencia en el DF del sitio, así como sus capacidades y el resto de los agentes pueden preguntarle al DF sobre ciertas capacidades para así conocer quienes pueden ser sus colaboradores.

Para el caso del prototipo de JITIK el Site Agent es el único que hace consultas al DF sobre las direcciones de todos los agentes registrados así como a sus capacidades básicas. En el Cuadro 4.2 se describe el rol del agente.

Rol	Directory Facilitator, responde consultas en la páginas amarillas del sistema
Responsabilidades	Administra la correspondencia entre agentes brindando servicios de registro y des-registro así como servicios de consulta
Colaboradores	Site Agent
Interfases externas	-
Relaciones	-
Experiencia	Servicios de FIPA
Interacciones	FIPA-ACL
Otros	Proporcionados por JADE

Cuadro 4.2: Rol del agente Directory Facilitator de JADE

## Site Agent

Site Agent opera como un broker para la comunicación entre los distintos agentes dentro del sitio y con otros sitios. Es quien se encarga de recibir la petición de los agentes, consultar en el DF de acuerdo a las capacidades de los agentes y enviar alguna solicitud o mensaje al agente o agentes que corresponda. Es el único agente que interactúa con agentes de otros sitios a través del Site Agent de los sitios externos. En el Cuadro 4.3 se describe el rol del agente de sitio para efectos del prototipo de JITIK.

Rol	Site Agent, brinda la interacción entre los distintos agentes del sistema
Responsabilidades	Se encarga del intercambio de mensajes entre agentes
Colaboradores	Site Agent, Ontology Agent, Persona Agent, Directory Facilitator, Agent Management Systems
Interfases externas	JitikDB
Relaciones	-
Experiencia	Resolución de problemas, administración de la interacción entre agentes
Interacciones	FIPA-ACL
Otros	Corre de forma persistente, es uno por sitio

Cuadro 4.3: Instancia del agente de sitio para el prototipo de JITIK

## Ontology Agent

En el caso del prototipo de JITIK se utilizan ontologías para representar las áreas del conocimiento y su jerarquía, así como el universo de usuarios y sus áreas de interés.

El Ontology Agent utiliza mecanismos de inferencia para brindar su servicio. Por ejemplo: Si A está interesado en X y Y es subclase de X entonces A está interesado en Y.

El Ontology Agent es utilizado dentro de la arquitectura de JITIK para dar el servicio de ontologías a los demás agentes. El Ontology Agent resuelve preguntas del tipo “¿A quién le interesa conocer esta información?” “¿A qué categoría corresponde esta área de conocimiento?”. En el Cuadro 4.4 se describe el rol del agente de ontología para efectos del prototipo de JITIK.

Rol	Ontology Agent, administra la ontología común para todos los agentes
Responsabilidades	Responde consultas sobre la ontología, perfil de usuarios y áreas de interés
Colaboradores	Site Agent, Personal Agent, Agent Management Systems
Interfases externas	JitikDB
Relaciones	-
Experiencia	Inferencia, resolución de problemas
Interacciones	FIPA-ACL
Otros	Corre de forma persistente, uno en cada sitio

Cuadro 4.4: Instancia del agente de ontología para el prototipo de JITIK

### Personal Agent

El Personal Agent se encarga de la interacción con los usuarios de la arquitectura. Es el único agente con el que el usuario tiene interacción.

Para efectos del prototipo el Personal Agent hay tres funciones principales que hace el Personal Agent:

- notifica a los usuarios de cambios efectuados en un sitio Web de su interés
- permite la captura de algún mensaje que el usuario quiera notificar a otros usuarios interesados en algún área de interés.
- notifica a los usuarios algún mensaje generado dentro de sus áreas de interés.

En el Cuadro 4.5 se describe el rol del agente personal para el prototipo de JITIK.

Rol	Personal Agent, brinda la interacción con los usuarios
Responsabilidades	Maneja la interfaz entre los usuarios y el resto de los agentes en el sistema
Colaboradores	Site Agent, Ontology Agent, Agent Management Systems
Interfases externas	Interfase gráfica, JitikDB
Relaciones	-
Experiencia	Administración de la interfase del usuario
Interacciones	FIPA-ACL
Otros	El usuario decide cuando activarlo

Cuadro 4.5: Instancia del agente personal para el prototipo de JITIK

## Monitor Agent

Para efectos del prototipo de JITIK se implementó un servicio de monitor de páginas Web, donde el agente tiene una lista de sitios Web de interés para los usuarios del sistema, contenidos en la tabla PaginasWeb de la base de datos y donde se indica el periodo de monitoreo para cada página.

El Monitor Agent se encarga de estar visitando periódicamente las páginas Web para consultar si ha habido algún cambio en esos sitios de interés y así notificar al usuario. En el Cuadro 4.6 se describe el rol del agente monitor para el prototipo de JITIK.

Rol	Monitor Agent, notifica de cambios en contenido de páginas Web
Responsabilidades	Revisa si el contenido de las páginas Web ha cambiado y lo notifica
Colaboradores	Site Agent, Agent Management Systems
Interfases externas	Internet, JitikDB
Relaciones	-
Experiencia	Resolución de problemas
Interacciones	FIPA-ACL
Otros	Corre de forma persistente

Cuadro 4.6: Instancia del agente monitor en el prototipo de JITIK

## Bridge Agent

Dentro del prototipo de JITIK se implementó un servicio de bridge o puente con un sistema de información externo como es la biblioteca digital del ITESM.

No fue posible conectar el prototipo al sistema real de la biblioteca, pero el servicio se simula a través de un mensaje generado por un usuario de la arquitectura. En el Cuadro 4.7 se describe el rol del agente puente para el prototipo de JITIK.

Rol	Bridge Agent, consulta los libros en el sistema de biblioteca
Responsabilidades	Consultar los libros nuevos relacionados de cierta área de conocimiento en la biblioteca digital del ITESM
Colaboradores	Site Agent, Agent Management Systems
Interfases externas	Sistema de biblioteca(función simulada en el prototipo), JitikDB
Relaciones	-
Experiencia	Resolución de problemas
Interacciones	FIPA-ACL
Otros	Corre de forma persistente

Cuadro 4.7: Instancia del agente bridge o puente para el prototipo de JITIK

La arquitectura de JITIK con los servicios proporcionados por JADE y la base de datos de cada sitio quedaría como se muestra en la Figura 4.3.

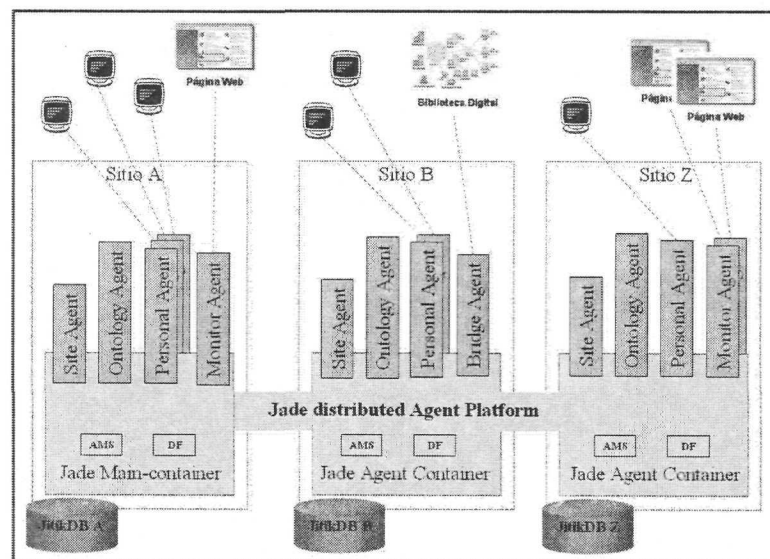


Figura 4.3: Arquitectura de JITIK integrada con plataforma de JADE

## 4.2. Servicios de JITIK

En el prototipo de JITIK se implementaron los servicios de ontologías para responder a preguntas sobre áreas de interés de los usuarios, un servicio de monitoreo de páginas Web, el servicio de bridge o puente con el sistema de biblioteca digital del ITESM y el servicio de mensajes, este último a su vez tiene varias modalidades. A continuación se especifica cada uno de los servicios.

### Servicio de Ontología

El servicio de ontologías es un servicio básico de la arquitectura. Permite relacionar áreas de conocimiento con áreas más generales. También brinda servicio sobre los intereses que tienen los usuarios en distintas áreas de conocimiento.

Este es un servicio que brinda soporte a otros agentes dentro del sistema y se ve reflejado en los servicios que se presentan más adelante en este capítulo:

- Servicio de Mensajes por Area, un Personal Agent que reciba información sobre cierta área de conocimiento utiliza el servicio de ontologías para conocer si esa área de interés le concierne en base a su perfil predefinido.
- Servicio Bridge con la Biblioteca Digital, el Site Agent recibe la notificación de que existe un nuevo libro en biblioteca. El Site Agent utiliza el servicio de ontologías para saber a que usuarios dentro del sistema concierne esta información y enviarles un mensaje.



## Servicios de Mensajería

Otro servicio implementado dentro del prototipo es el de envío de mensajes que tiene tres modalidades:

### a) Servicio de Mensajería por Grupo

Los mensajes por grupo sirven para ayudar en esquemas de trabajo colaborativo. Algún mensaje generado por un usuario o sistema y que concierne a cierta grupo de trabajo. El usuario o sistema genera el mensaje y éste será distribuido a los usuarios dentro grupo de trabajo independientemente del sitio al que pertenezcan.

Un ejemplo de aplicación de este servicio sería dentro una empresa de manufactura que produce empaques de cartón y que ha creado un equipo de trabajo multidisciplinario para lograr alguna certificación de la norma ISO(International Organization for Standardization).

Los miembros de grupo podrían utilizar el servicio de JITIK de mensajes por grupo para notificar fechas, avances, revisión y cambios dentro del plan de trabajo del proceso de certificación a los miembros del grupo. En la Figura 4.4 se muestra la secuencia de pasos para el envío de mensajes en grupo.

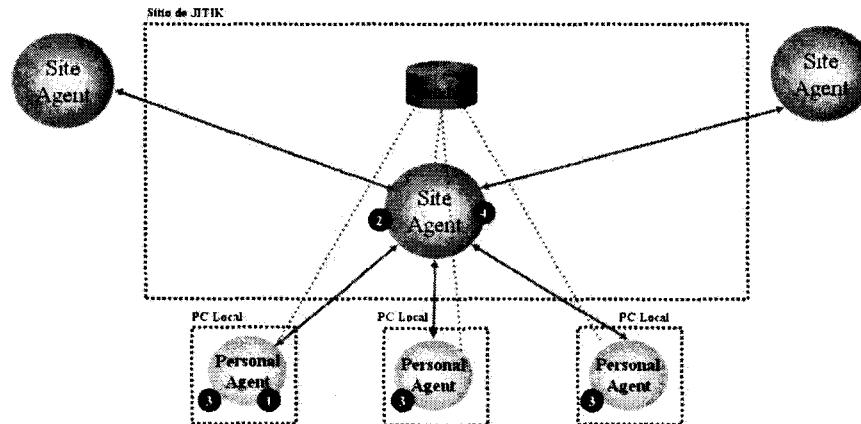


Figura 4.4: Secuencia de pasos en el servicio de mensajería por grupo

1. Genera un mensaje para un grupo determinado, se lo envía al Site Agent.

2. El SiteAgent, en base a la información que conoce a cerca de los grupos contenida en la tabla UsuariosxGrupo, le notifica a los PersonalAgent locales interesados y envía el mensaje a los demás sitios. Si no puede entregar algún mensaje, lo guardará en la tabla Mensajes para posterior entrega.
3. Cuando un PersonalAgent inicia, verifica si tiene algún mensaje pendiente.
4. El SiteAgent tiene un proceso constante para reenviar los mensajes que no se hayan podido entregar a otros sitios.

En la Figura 4.5 se muestra el diagrama de secuencia UML para el servicio de Mensajería por Grupo, donde el Usuario A y el Usuario C son miembros del grupo y el usuario B es el que envía la notificación del mensaje. Por otro lado SiteAgentX representa a los SiteAgents de otros sitios externos.

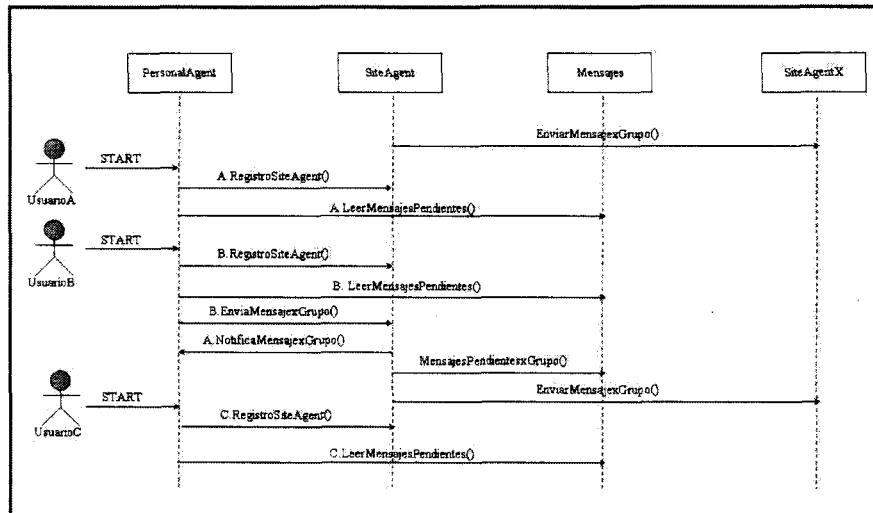


Figura 4.5: Diagrama UML para el servicio de mensajería por grupos

a) Servicio de Mensajería por Área

Este servicio se concentra en notificar algún mensaje generado por un usuario y que concierne a cierta área de conocimiento. El usuario genera el mensaje y éste será distribuido a los usuarios dentro del sitio interesado en esa área de conocimiento así como a los usuarios de otros sitios igualmente interesados.

Un ejemplo de la utilidad de este servicio es en el área académica, un profesor investigador que publica un nuevo artículo sobre su última aportación al área de ingeniería genética notifica a JITIK sobre el nuevo artículo y el resto de los investigadores o interesado en ingeniería genética pueden tener conocimiento sobre esta nueva aportación al área de investigación de forma inmediata. En la Figura 4.6 se muestra la secuencia de pasos para el envío de mensajes por área.

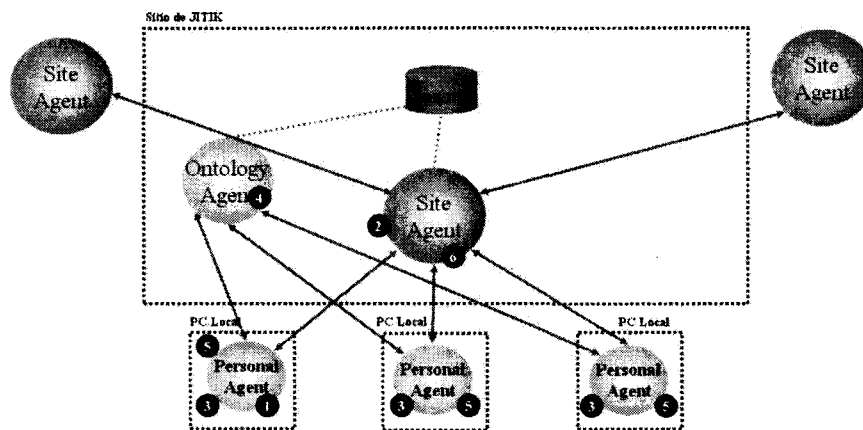


Figura 4.6: Secuencia de pasos en el servicio de mensajería por áreas

1. Genera un mensaje para un área determinada, se lo envía al Site Agent.
2. El SiteAgent le notifica del mensaje a todos los PersonalAgent locales y a los demás SiteAgents del sistema. Si no puede entregar algún mensaje, lo guardará en la tabla Mensajes para posterior entrega.
3. Cada PersonalAgent se comunica con el OntologyAgent para, en base a la información de las áreas de interés, le resuelva si el mensaje le concierne o no.
4. El OntologyAgent tiene un mecanismo de inferencia para resolver información relacionada con la jerarquía de información y usuarios.

5. Cuando un PersonalAgent inicia, verifica si tiene algún mensaje pendiente.
6. El SiteAgent tiene un proceso constante para reenviar los mensajes que no se hayan podido entregar a otros sites.

Enseguida se muestra en la Figura 4.7 el diagrama de secuencia UML para el servicio de Mensajes por Area de Interés, donde el Usuario A y el Usuario C están interesados en cierta área y el usuario B es el que envía la notificación del mensaje relacionado con el área. Por otro lado SiteAgentX representa a los SiteAgents de otros sitios externos.

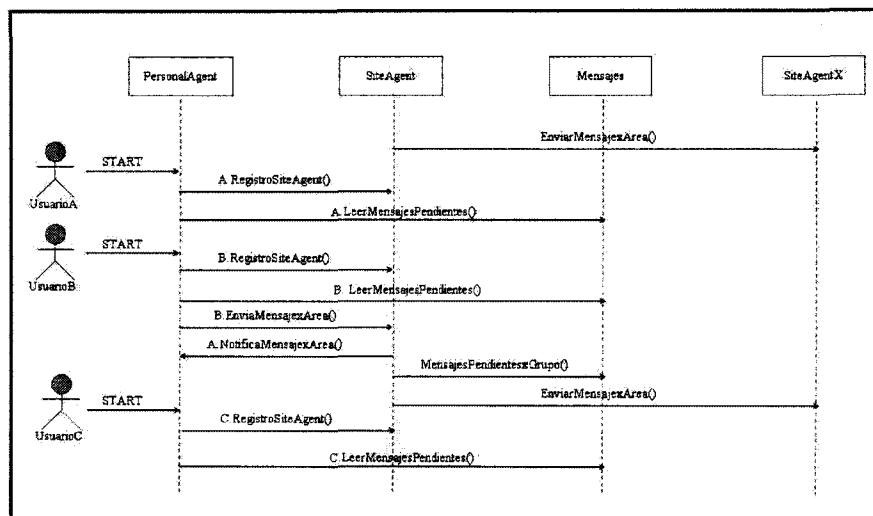


Figura 4.7: Diagrama UML para el servicio de mensajería por área de interés

*a) Servicio de Mensajería por Usuario*

El servicio de mensajes por usuario es la versión simplificada de mensaje por grupos, aquí el mensaje únicamente se dirige a un usuario que puede estar en el sitio local o en otro sitio externo.

Los mensajes podrían venir de la integración con otro sistema externo o de algún usuario. Para efectos del prototipo los mensajes son enviados de usuario a usuario.

Los Site Agents colaboran para localizar al usuario y entregarle el mensaje, si el usuario no se encuentra activo recibirá el mensaje la próxima vez que se conecte al sitio. La Figura 4.8 presenta la secuencia de pasos para el envío de mensajes para un usuario en particular.

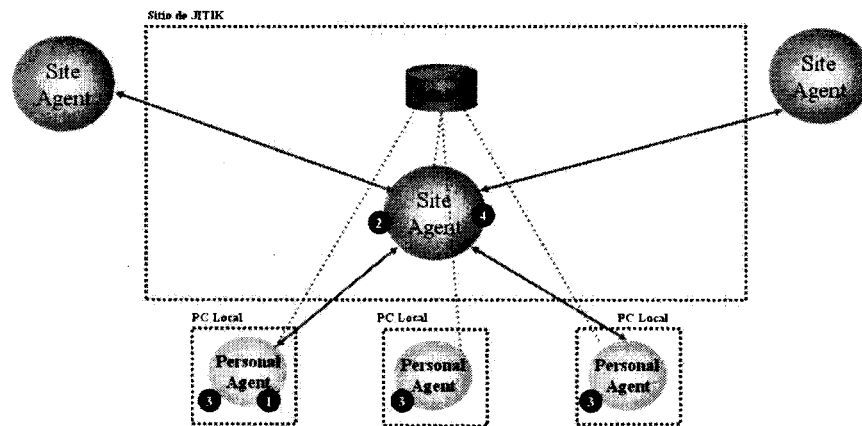


Figura 4.8: Secuencia de pasos en el envío de mensajería por usuario

- 1. Genera un mensaje para otro usuario determinado, se lo envía al Site Agent.*
- 2. El SiteAgent, si encuentra que el usuario es local, le envía el mensaje, si no, envía el mensaje a los demás sitios. Si no puede entregar algún mensaje, lo guardará en la tabla Mensajes para posterior entrega.*
- 3. Cuando un PersonalAgent inicia, verifica si tiene algún mensaje pendiente.*
- 4. El SiteAgent tiene un proceso constante para reenviar los mensajes que no se hayan podido entregar a otros sitios.*

A continuación se muestra en la Figura 4.9 el diagrama de secuencia UML para el servicio de Mensajes por Usuario, donde el Usuario B le manda un mensaje al Usuario A a través del SiteAgent.

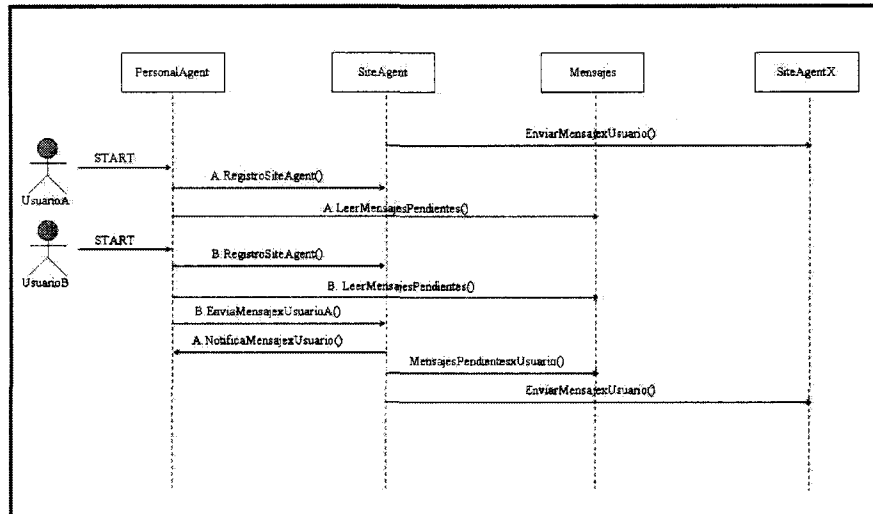


Figura 4.9: Diagrama UML para el servicio de mensajería por usuario

## Servicio de Monitoreo de Páginas Web

El primero de los servicios implementados en el prototipo de JITIK es el monitoreo de páginas Web. Este servicio tiene como objetivo permitir que los usuarios tengan conocimiento de que alguna página de su interés ha cambiado su contenido sin necesidad de que el usuario tenga que ir visitarla periódicamente.

Un sitio de noticias, el sitio Web de la competencia o algún sitio donde se esté reportando las últimas amenazas en virus informáticos son el tipo de sitios que un usuario estaría interesado en monitorear. A continuación en la Figura 4.10 se describe la forma en que este servicio se lleva a cabo.

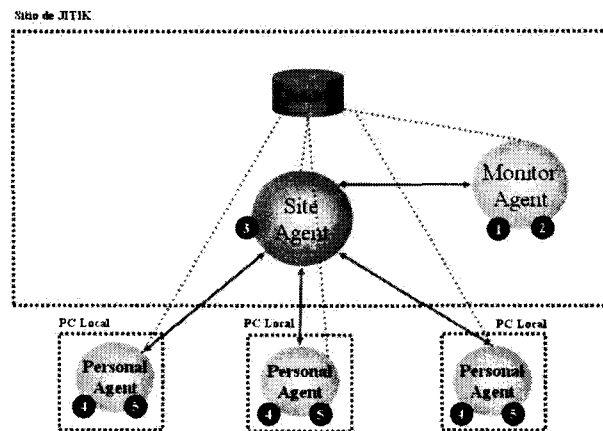


Figura 4.10: Secuencia de pasos para el servicio de monitoreo de páginas Web

1. Es un proceso que se encarga de disparar la verificación de cambio de páginas en base a su periodicidad definida en la tabla *PaginasWeb*.
2. Por cada sitio Web, verifica si el site ha cambiado de acuerdo a la fecha o bien, el contenido. Actualiza la base de datos con el resultado obtenido y le notifica al *SiteAgent*.
3. Le notifica a todos los *PersonalAgent* de un cambio en algún site de internet. Si algún *PersonalAgent* no se encuentra en línea, guardará el mensaje en la tabla *Mensajes* para posterior entrega.
4. Cuando un *PersonalAgent* inicia, verifica si tiene algún mensaje pendiente en la tabla de *Mensajes*.

5. El *PersonalAgent*, en base a los intereses del usuario (definidos en la tabla *PaginasWebxUsuario*) filtra si los mensajes son de su interés.

En la Figura se presenta el diagrama de secuencia UML para el servicio de Monitoreo de Páginas Web, donde se refleja como proactivamente el *MonitorAgent* está enviando notificaciones a los usuarios interesado.

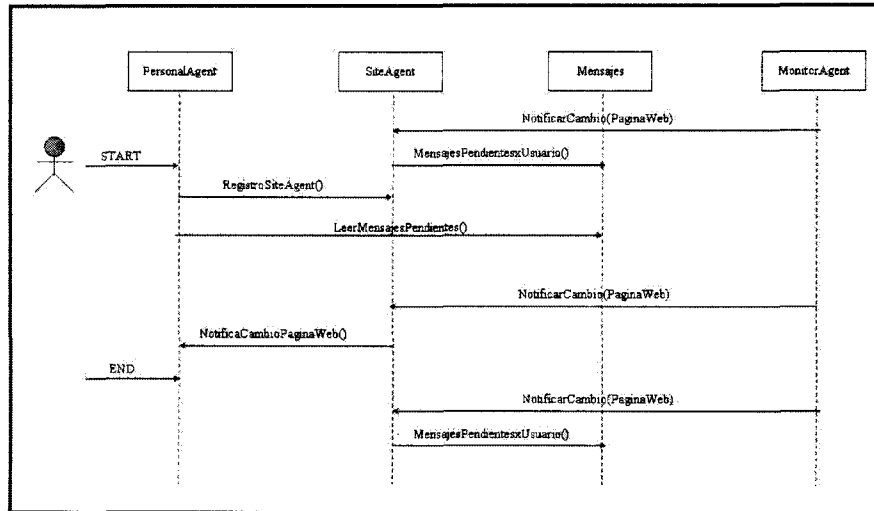


Figura 4.11: Diagrama UML para el servicio de monitoreo de páginas Web



### Servicio de Bridge con un sistema externo (Biblioteca Digital)

El servicio bridge o puente con un sistema externo se realizó simulando la integración con el sistema de biblioteca digital del ITESM.

La consulta sobre los nuevos libros la dispara uno de los usuarios de la arquitectura y se despliega un mensaje a todos los interesados en esa área de conocimiento. A continuación en la Figura 4.12 se describe la forma en que este servicio se lleva a cabo.

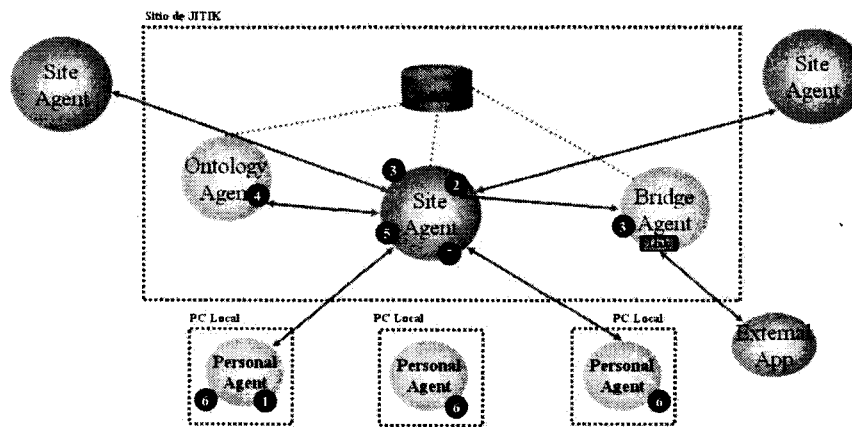


Figura 4.12: Secuencia de pasos en el servicio de bridge con el sistema de biblioteca

1. Un usuario del sistema solicita la consulta de nuevos libros en la biblioteca digital al Site Agent sobre cierta área de conocimiento.
2. El Site Agent transfiere la petición al Bridge Agent.
3. El Bridge Agent realiza la consulta y regresa la respuesta al Site Agent.
4. El SiteAgent consulta con el Ontology Agent sobre los usuarios interesados en esa área de conocimiento. El OntologyAgent tiene un mecanismo de inferencia para resolver información relacionada con la jerarquía de información y usuarios.
5. El Site Agent entrega el mensaje a los PersonalAgent interesado en el área de conocimiento en cuestión y a otros Site Agents. Si no puede entregar algún mensaje, lo guardará en la base de datos para posterior entrega.
6. Cuando un PersonalAgent inicia, verifica si tiene algún mensaje pendiente.

7. El SiteAgent tiene un proceso constante para estar reenviando los mensajes que no se hayan podido entregar a otros sites.

En la Figura 4.13 se presenta el diagrama UML para representar el servicio de bridge o puente hacia un sistema externo, donde se muestra como el Usuario B dispara la consulta al sistema de biblioteca y al recibir la respuesta el Site Agent notifica a los usuarios interesados en el área al que el libro pertenece.

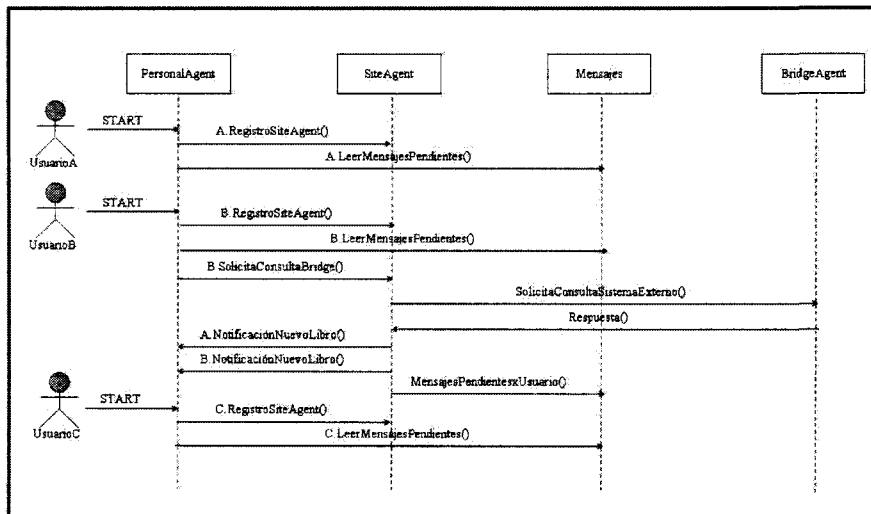


Figura 4.13: Diagrama UML para el servicio bridge con el sistema externo de biblioteca

### 4.3. Pruebas

Con el objetivo de validar lo planteado en el capítulo anterior es necesario realizar pruebas de los servicios implementados.

Para la realización de las pruebas se tomaron los escenarios planteados en el Capítulo 3 y para cada uno de ellos se seleccionó el servicio de JITIK que mejor cumple con el requerimiento de información o conocimiento que se desea cubrir.

En el Apéndice B se presentan los valores contenidos en la base de datos y sobre los cuales se realizaron las pruebas.

### 4.3.1. ESCENARIO DE PRUEBA A - Aviso de noticias relacionadas con la competencia

#### Objetivo

Validar el funcionamiento de JITIK para el servicio de monitoreo de páginas Web en base a los intereses de un usuario del sistema.

#### Contexto de la prueba

David Flores es usuario de JITIK y es un profesionalista en el área de informática y trabaja como Gerente de Ventas en la empresa IronPort Systems que se dedica a comercializar productos de seguridad en internet. A David le interesa conocer las noticias que se publican en los sitios de su competencia. Seleccionamos los sitios web de las empresas WebSense, TrendMicro y BlueCoat y dentro de los sitios revisamos aquellas páginas que reportan noticias o notas de prensa relacionadas con la empresa. Se estableció una periodicidad de un día para monitorearlas ya que se conoce que estos sitios se actualizan en periodos de al menos un día.

#### Resultado esperado

Con esta prueba el resultado que se espera es que cada vez que suceda un cambio en los sitios de noticias de las páginas Web que se están monitoreando con JITIK, David Flores reciba una notificación sobre el cambio en el sitio Web.

#### Prueba

El Monitor Agent periódicamente está revisando cuando es el momento de verificar un sitio Web de acuerdo a la frecuencia de monitoreo que se le haya definido. Los sitios Web que se definieron se muestran en el Cuadro 4.8.

Sitio Web	Periodicidad
<a href="http://www.websense.com/global/en/PressRoom/">http://www.websense.com/global/en/PressRoom/</a>	24 horas
<a href="http://trendmicro.mediaroom.com/">http://trendmicro.mediaroom.com/</a>	24 horas
<a href="http://www.symantec.com/about/news/release/">http://www.symantec.com/about/news/release/</a>	24 horas
<a href="http://www.altonivel.com.mx/newsletter.html">http://www.altonivel.com.mx/newsletter.html</a>	24 horas
<a href="http://www.bluecoat.com/news/pressreleases">http://www.bluecoat.com/news/pressreleases</a>	24 horas
<a href="http://www.ironport.com/toc/">http://www.ironport.com/toc/</a>	24 horas
<a href="http://www.bsecure.com.mx/">http://www.bsecure.com.mx/</a>	24 horas

Cuadro 4.8: Sitios Web utilizados para prueba de monitoreo

Cuando se llega el momento de ser revisada, el MonitorAgent verifica el estado del sitio y si ha cambiado notifica al SiteAgent sobre el cambio.

El SiteAgent notifica a todos los PersonalAgent que se encuentren en línea sobre el cambio en el sitio Web y deja un mensaje pendiente en la tabla de mensajes para aquellos PersonalAgent que no se encuentren conectados en ese momento.

Los PersonalAgent conectados en ese momento reciben el mensaje y validan en la tabla PaginasWebxUsuario si el usuario tiene registrado interés en el sitio que le están notificando.

Los PersonalAgent que no estén conectados en ese momento, validarán si existen mensajes pendientes para ellos la próxima vez que se conecten y al encontrar mensajes con notificación de cambios en páginas Web, igualmente valida en la base de datos si el usuario tiene registrado interés en el sitio que le están informando.

## **Resultado**

El usuario David Flores recibe notificación de los siguientes sitios Web:

- <http://www.websense.com/global/en/PressRoom/>
- <http://trendmicro.mediaroom.com/>
- <http://www.bluecoat.com/news/pressreleases>

David, si lo desea, puede ir a consultar las páginas Web y enterarse de las últimas noticias respecto a su competencia con la certeza de que hay nueva información. Enseguida en las Figuras 4.14, 4.15 y 4.16 se muestran los sitios de noticias seleccionados para la prueba.

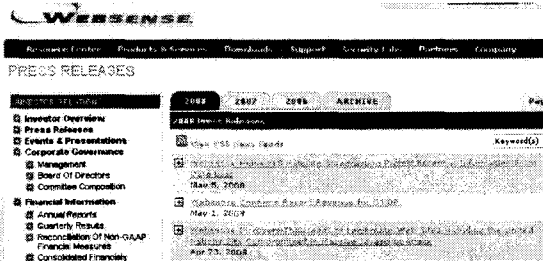


Figura 4.14: Sitio de noticias de WebSense



Figura 4.15: Sitio de noticias de TrendMicro

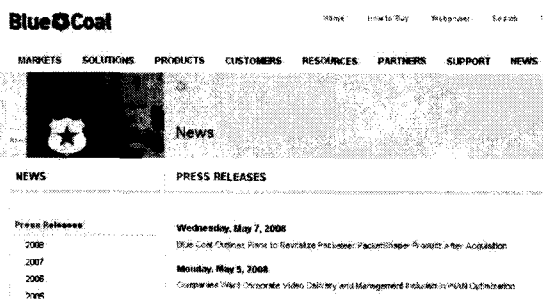


Figura 4.16: Sitio de noticias de BlueCoat

### 4.3.2. ESCENARIO DE PRUEBA B - Coleccionista de modelos de autos a escala

#### Objetivo

Validar el funcionamiento de JITIK para el servicio de envío de mensajes por grupo.

#### Contexto de la prueba

Para realizar la prueba se creó un grupo llamado “Subastas Autos a Escala” donde hay un grupo de personas que entre sus pasatiempos se encuentra coleccionar modelos de autos a escala, los cuales adquiere generalmente participando en subastas o comprándoselos a amigos que comparten el mismo interés. En el grupo se registraron los usuarios de JITIK Hugo Gómez y Eduardo Sotelo.

Otro usuario de JITIK, Carlos Cantú que sabe de la existencia del grupo y el interés de los usuario envía un aviso al grupo “Subastas Autos a Escala” informando sobre la subasta de un auto a escala hecho en vidrio murano que encontró navegando en eBay.

#### Resultado esperado

De acuerdo a las personas que conforman el grupo, los usuarios Hugo Gómez y Eduardo Sotelo reciben el mensaje ***“Está en subasta en eBay el auto NASCAR de colección en vidrio Murano, precio \$79 USD”***.

#### Prueba

El usuario Carlos Cantú consulta el sitio de eBay e identifica el auto miniatura que se presenta en la Figura 4.17 en subasta.

Carlos Cantú considera el dato interesante y desea dárselo a conocer las personas interesadas en este tema registradas en JITIK, así que genera el mensaje ***“Está en subasta en eBay el auto NASCAR de colección en vidrio Murano, precio \$79 USD”*** y lo envía al SiteAgent indicando que va dirigido al grupo “Subastas Autos a Escala”.

El SiteAgent recibe el mensaje y consulta los PersonalAgent locales que pertenecen a ese grupo. Si está en línea le entrega el mensaje, si no deja un mensaje pendiente en la tabla de Mensajes.

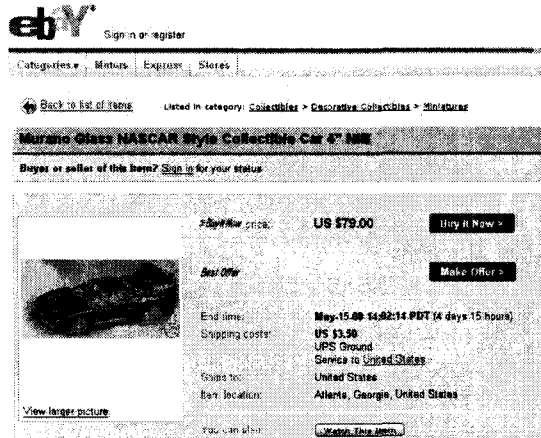


Figura 4.17: Auto minuatatura en subasta en eBay

El SiteAgent notifica a otros SiteAgent activos en ese momento sobre el mensaje para que sea entregado a los PersonalAgent de ese sitio. Si hay algún SiteAgent que no esté activo en ese momento, el SiteAgent reintentará el envío del mensaje.

El PersonalAgent recibe el mensaje si está en línea y le notifica al usuario.

Los PersonalAgent no conectados en ese momento, la próxima vez que se conecten irán a validad si tienen mensajes pendientes y al encontrar un mensaje de grupo le notificarán al usuario.

## Resultado

Los usuarios interesados en coleccionar modelos de autos a escala podrán conocer información relacionada que otros usuarios de JITIK notifiquen.

Los usuarios, si lo desean, pueden conectarse al sitio de eBay para participar en la subasta el auto a escala.

### 4.3.3. ESCENARIO DE PRUEBA C - Nuevos libros en biblioteca

#### Objetivo

Validar el funcionamiento de JITIK para el servicio de bridge con el sistema de biblioteca digital del ITESM.

#### Contexto de la prueba

Para realizar la prueba se tomaron como áreas de interés la clasificación hecha por ACM (Association for Computing Machinery) en el apartado I.2 que se refiere a Inteligencia Artificial. Ver Apéndice C. [18]

Carlos Cantú, usuario de JITIK, es profesor de la maestría de Sistemas Inteligentes en el ITESM y le interesan los temas relacionados con Inteligencia Artificial Distribuida por lo que tiene registrado en perfil de usuario recibir todas las notificaciones de nuevos libros en biblioteca relacionados con el área.

Susana Trujillo es alumna del ITESM en la maestría de Sistemas Inteligentes y es usuaria de JITIK y a través de la plataforma solicita información sobre las nuevas adquisiciones de biblioteca relacionadas con el área de Inteligencia Artificial.

#### Resultado esperado

Los usuarios que hayan registrado interés en alguna área relacionada con Inteligencia Artificial recibirán la notificación de nuevas adquisiciones en biblioteca cuando algún usuario del sistema solicite la consulta.

#### Prueba

Susana Trujillo solicita la consulta al Site Agent sobre libros nuevos dentro del área de Inteligencia Artificial.

El Site Agent a su vez, transfiere la petición al Bridge Agent.

El Bridge Agent resuelve la consulta y encuentra el nuevo libro que se muestra en la Figura 4.18 y contesta con el mensaje ***“NUEVO LIBRO EN BIBLIOTECA: [QA 76.76.I58 A72 2007] Architectural design of multi-agent systems: technologies and techniques Autor Hong Lin, 01-Jan-2007”*** al SiteAgent dirigido



al área de interés de “I.2.11 Distributed Artificial Intelligence”.

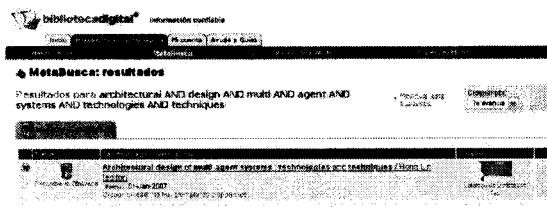


Figura 4.18: Libro nuevo en biblioteca digital del ITESM

El SiteAgent recibe el mensaje y consulta con el Ontology Agent sobre los usuarios de la arquitectura interesados en esa área de conocimiento “I.2.11 Distributed Artificial Intelligence”.

El OntologyAgent realiza un proceso de inferencia para conocer con qué otras áreas está relacionada el área de “Distributed Artificial Intelligence” y encuentra que está relacionada con el área general de “I.2 Artificial Intelligence”. Resuelve la lista de usuarios interesados en esas áreas y se la regresa al Site Agent.

El Site Agent envía la notificación a los PersonalAgent interesados y locales que se encuentran en línea en ese momento. Si no está en línea deja un mensaje pendiente en la tabla de Mensajes.

El SiteAgent notifica a otros SiteAgent activos en ese momento sobre el mensaje para que sea entregado a los PersonalAgent de ese sitio. Si hay algún SiteAgent que no esté activo en ese momento, el SiteAgent reintentará el envío del mensaje. Los PersonalAgent no conectados en ese momento, la próxima vez que se conecten a JITIK revisarán si tienen mensajes pendientes.

## Resultado

Como resultado de la solicitud de Susana Trujillo sobre consultas en el área de Inteligencia Artificial los usuarios que reciben la notificación “**NUEVO LIBRO EN BIBLIOTECA: [QA76.76.I58 A72 2007] Architectural design of multi-agent systems: technologies and techniques Autor Hong Lin, 01-Jan-2007**” son: David Flores, Carlos Cantú, Erika Buenrostro, Susana Trujillo y Hugo Gómez.

Los usuarios pueden consultar, si así lo desean, la nueva referencia bibliográfica disponible en biblioteca relacionada con el área de Inteligencia Artificial.

## Capítulo 5

### Conclusiones

En este capítulo final se presentan las conclusiones obtenidas así como las aportaciones de la tesis presentada. Adicionalmente se mencionan trabajos relacionados y cual sería una posible extensión de la misma en trabajos futuros.

#### 5.1. Conclusiones

En la presente tesis se propuso validar que una arquitectura de sistemas multi-agentes que permita la interacción de diversos repositorios de conocimiento puede ayudar en los servicios de distribución de conocimiento.

Para lo anterior fue necesario realizar un conjunto de tareas, comenzando por investigar donde se encuentra almacenada la información y el conocimiento computacionalmente. Se encontró que Internet es una fuente rica de información y conocimiento, al igual que las bibliotecas digitales y las memorias organizacionales.

Bajo ese esquema se diseñó la arquitectura de JITIK basada en sistemas multi-agentes que permitiera a usuarios interactuar con algunos de los principales repositorios de información y conocimiento.

Para cada fuente de información se diseñaron roles de agentes para llevar a cabo las funciones de localización y distribución del conocimiento o información. Los roles definidos dentro de la arquitectura son: Site Agent, Ontology Agent, Personal Agent, Monitor Agent y Bridge Agent.

Se investigaron las herramientas disponibles para desarrollar un prototipo de la arquitectura, roles y servicios planteados. La herramienta seleccionada fue JADE.

Se implementó un prototipo de la arquitectura multi-agente que incluye los roles

de agentes y varios servicios de distribución. Los servicios implementados en el prototipo fueron: Servicio de Ontologías, Servicio de Monitoreo de Páginas Web, Servicio de Bridge con una Biblioteca Digital y una serie de Servicios de Mensajes.

Finalmente se diseñaron escenarios de prueba para validar el funcionamiento de la arquitectura, obteniendo resultados esperados.

Con base en el trabajo desarrollado podemos concluir que efectivamente las tecnologías de sistemas multi-agentes son una excelente alternativa para la implementación de mecanismos de distribución de información y conocimiento, en contraste con tecnologías informáticas convencionales.

A diferencia de las tecnologías informáticas convencionales, las tecnologías de sistemas multi-agentes tienen características de autonomía, cooperación, procesos de larga vida, etc. que hacen que los agentes actúen como “personas virtuales” dentro de una organización.

El concepto de “personas virtuales” en arquitecturas como JITIK puede extenderse al de una “organización virtual” donde el componente Site Agent por ejemplo puede jugar el papel de gerente de alguna división dentro de la organización. Los Personal Agent pueden tomar el rol de sus colegas y el Monitor Agent o Bridge Agent realizarían tareas que comúnmente son asignadas a sus subordinados.

Las ventajas de esa perspectiva son entre otras que los agentes no descansan, no se les olvidan las cosas y no tienen opinión personal de otros agentes, que los lleven a hacer acciones en contra de los intereses de la organización.

En ese sentido y a pesar de que no se demostró en esta tesis creemos que la administración de conocimiento puede verse beneficiada ampliamente en el uso de tecnologías implementadas como JITIK.

## **5.2. Aportaciones**

La principal aportación de la tesis es demostrar que la tecnología de Sistemas Multi-agentes puede resultar una herramienta útil en apoyo a la distribución del conocimiento de manera oportuna y de acuerdo a intereses predefinidos para los usuarios.

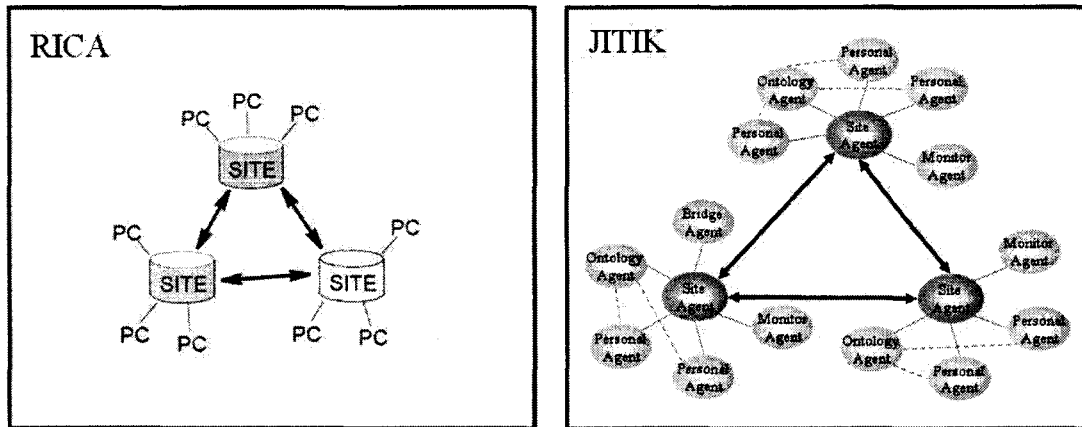


Figura 5.1: Comparación entre arquitecturas de RICA y JITIK

Recordando la arquitectura anterior a la tesis establecida en RICA que se muestra en la Figura 5.1 a la izquierda y contrastándola con la arquitectura propuesta en la tesis para JITIK que se muestra en la Figura 5.1 a la derecha.

Si comparamos ambas arquitecturas tenemos que una arquitectura como JITIK nos brinda una mayor flexibilidad para extenderla ya que los agentes se encuentran claramente definidos.

En JITIK la definición de los roles de agentes se han definido de manera abstracta y se pueden crear instancias de ellos en base a las funciones o tareas que se le encomienden. Por ejemplo es posible definir varios tipos de Monitor Agent, Bridge Agent o Personal Agent.

Por otro lado la implementación en JITIK es más robusta ya que en JITIK el Site Agent no es necesario que sea ejecutado 24 horas ni en servidores muy robustos, tal y como se define para el componente similar en RICA. En JITIK el Site Agent puede correr en cualquier servidor y si por alguna razón es interrumpido su funcionamiento tiene mecanismos para recuperar mensajes que se hayan generado incluso a nivel de Sitio.

### 5.3. Trabajos relacionados

Para presentar los trabajos relacionados se decidió organizarlos en base a los procesos básicos de la Administración del Conocimiento (AC) que son: creación, aseguramiento, distribución y combinación [5].

Enseguida en el Cuadro 5.1 se presenta la relación entre diversas arquitecturas con los procesos básicos del la AC y más adelantes se presenta una breve descripción de las arquitecturas:

Arquitecturas de SMA aplicadas a AC	Crear	Asegurar	Distribuir	Combinar
Arquitectura de un sistema AC basado en agentes			x	
Soporte de Agentes Inteligentes para Sistemas de AC	x	x	x	x
CoMMA -Corporate Memory Management through Agents	x	x		
DIAMS - Distributed Multi-agent System for Collaborative Information Management and Sharing			x	
FRODO - Framework for Distributed Organisational Memories		x	x	
KRAFT - Knowledge Reuse and Fusion/Transformation				x
Proyecto Campiello	x	x	x	
AMKM - Agent-mediated Knowledge Management			x	x

Cuadro 5.1: Trabajos relacionados de agentes inteligentes con AC

### Arquitectura de un Sistema de AC basado en Agentes

Este trabajo [20] se basa en las siguientes consideraciones:

- en una organización existen diferentes entidades de conocimiento (usuarios, equipos, comunidades, etc.)
- en una organización se identifican 3 tipos de conocimiento, que son el técnico (de naturaleza explícita), el conocimiento o experiencia en procesos (de naturaleza tácita), y el conocimiento sobre cadenas de relaciones (también de naturaleza explícita).

Con base en lo anterior, se propone una arquitectura que consiste de un conjunto de sociedades de agentes de dos tipos. Por un lado encontramos las que representan las entidades de conocimiento (usuarios, equipos y comunidades), y por otro lado las que administran los objetos de información y conocimiento.

Una ontología es compartida por todos los miembros de la organización y contiene conocimiento de los tres tipos descritos con anterioridad; una sociedad de servicios registra las fuentes disponibles y procesa las peticiones de servicios.

La sociedad de usuarios consta de 3 tipos diferentes de agentes: un agente mantiene el perfil de intereses del usuario, otro se encarga de determinar lo que puede ser relevante para el usuario, y el último solicita la información a la sociedad de servicios; la sociedad de equipos consiste en dos agentes que administran el conocimiento técnico y sobre procesos; finalmente la sociedad de comunidades contiene dos agentes que proveen el conocimiento sobre las cadenas de relaciones, y controlan los intercambios de información entre los miembros de un equipo.

### **Soporte de Agentes Inteligentes para Sistemas de AC**

Este trabajo [21] se fundamenta en el hecho de que un sistema que implemente los cuatro procesos de manejo del conocimiento permitirá a las organizaciones resolver aspectos como la solución de problemas, el aprendizaje dinámico, la planeación estratégica y la toma de decisiones.

En la arquitectura propuesta se distingue un componente que administra las comunicaciones entre los agentes y el mundo exterior. El componente central contiene un sistema de flujo de conocimiento que provee facilidades como la búsqueda de conocimiento o la recepción y procesamiento de conocimiento, además de un coordinador que funge como mediador para llevar a cabo negociaciones entre agentes. Por otro lado se cuenta también con un repositorio de conocimiento.

Por último, se distinguen dos tipos de agentes inteligentes de aplicación: los genéricos y los específicos. Los agentes genéricos incluyen al agente de búsqueda de conocimiento, y a los agentes de recepción y procesamiento de conocimiento. Los agentes específicos incluyen al agente distribuidor de conocimiento, y a los agentes negociador y de monitoreo de conocimiento.

### **CoMMA**

CoMMA (por “Corporate Memory Management through Agents”) [22] es un proyecto que tiene como objetivo implementar y probar un marco de administración de una memoria corporativa integrando diversas tecnologías como SMA, modelación de conocimiento, XML, recuperación de información y aprendizaje automático.

La solución propuesta está basada en una arquitectura multiagente de agentes cooperativos. El sistema multiagente tendrá interacción con la memoria corporativa, materializada en documentos XML, y con modelos de conocimiento representados por una ontología en RDF. En el SMA, cada usuario tendrá su propio agente, a los cuales se agregan agentes que representan grupos de interés y agentes de soporte a la autoría

de documentos.

Los agentes utilizan mecanismos de inferencia para explotar las ontologías, y técnicas de aprendizaje automático son empleadas para mejorar el rendimiento de los agentes de modo que se adapten a los usuarios y al contexto en que son ejecutados. La arquitectura del SMA de CoMMA se encuentra implementada en JADE.

## **DIAMS**

DIAMS (por “Distributed Multi-agent System for Collaborative Information Management and Sharing”) [23] es un sistema que ayuda a los usuarios a acceder, administrar, compartir e intercambiar información relevante para ellos en la Web. El esquema de colaboración de los agentes en DIAMS incluye agentes personales que se encargan de presentar, organizar y administrar las colecciones de información de los usuarios.

Esos agentes interactúan con otros dos tipos de agentes, que son los agentes de asociación y los agentes de conocimiento. Los primeros facilitan la colaboración entre agentes personales, pues se ocupan de obtener conocimiento de otros usuarios, y de comunicar los que presentan intereses más cercanos; los segundos contienen conocimiento especializado que ponen al servicio de los agentes personales.

## **FRODO**

FRODO (por “Framework for Distributed Organisational Memories”) [24] se enfoca en el desarrollo de métodos y herramientas para construir y mantener memorias organizacionales distribuidas en un ambiente de empresas reales.

Su objetivo es definir una capa intermedia para las memorias organizacionales, donde se insertan distintos tipos de agentes ontológicos, que conectan el nivel de fuentes de conocimiento con el nivel de aplicaciones.

El producto definido soporta además dos tipos de escalabilidad. La escalabilidad vertical permite agregar elementos en cada nivel del modelo, por ejemplo nuevos procesos en el nivel de aplicación; la escalabilidad horizontal permite que diferentes sistemas de memorias organizacionales interoperen para responder a consultas de manera cooperativa.

Eso se traduce entonces en la existencia de dos tipos de agentes: agentes ontológicos de dominio, encargados de asegurar la escalabilidad vertical, y agentes ontológicos distribuidos, encargados de asegurar la escalabilidad horizontal.

## **KRAFT**

KRAFT (por “Knowledge Reuse and Fusion/Transformation”) [25] soporta la fusión de conocimiento de múltiples fuentes heterogéneas y distribuidas.

KRAFT tiene una arquitectura adecuada para trabajar en organizaciones virtuales donde los miembros intercambian información en forma de restricciones expresadas con base en un modelo de objetos, lo cual lo hace ideal para ser usado en problemas de diseño de configuraciones.

La arquitectura de KRAFT está basada en agentes de cuatro tipos: los agentes de usuario, los agentes de envoltura (“wrappers”), los facilitadores y los mediadores. Los agentes de envoltura actúan como intermediarios (“proxies”) para fuentes externas de conocimiento, y también sirven como puntos de entrada al sistema para los usuarios.

Los agentes mediadores son los agentes que procesan el conocimiento al interior del sistema, y se ocupan de tareas como filtrado, ordenamiento y fusión del conocimiento obtenido de otros agentes. Finalmente los agentes facilitadores realizan funciones de asociación que permiten a los agentes estar enterados de lo que sucede y comunicarse en consecuencia.

## **Campiello**

El proyecto Campiello [26] utiliza el filtraje colaborativo como un elemento de servicio de AC. En este caso, el filtraje colaborativo se conceptualiza como la búsqueda reactiva y proactiva de información relevante en un contexto donde se tienen grandes cantidades de información.

Se proponen tres módulos para llevar a cabo esa búsqueda: el recomendador, el de búsqueda y un espacio compartido de información. El primero, como su nombre lo indica, realiza recomendaciones personalizadas sobre calificaciones que recibe de la información. El segundo realiza búsquedas sobre consultas específicas, y consiste básicamente de un motor de recuperación de información, que puede ser visto como un conjunto de agentes estructurados jerárquicamente. El espacio compartido de información contiene información sobre los usuarios, sus perfiles de interés y sobre mapas de las comunidades de trabajo. De estos módulos se desprenden varios agentes, tales como agentes para almacenar y administrar datos, agentes de envoltura y de búsqueda, agentes de generación de información y agentes recomendadores.



## AMKM

AMKM (por “Agent-mediated Knowledge Management”) [27] es un marco que usa conceptos de agentes para analizar y modelar organizaciones y sus necesidades de conocimiento. Cuenta con un conjunto de agentes responsables de hacer corresponder la oferta y demanda de conocimiento en una organización, tomando en cuenta las preferencias y necesidades de los usuarios, así como las necesidades de conocimiento de las tareas que ellos realizan.

Desde el punto de vista organizacional, AMKM propone una sociedad de agentes consistente de tres modelos interrelacionados: el modelo organizacional que describe el comportamiento deseado de una sociedad de agentes, el modelo social que asigna roles organizacionales a agentes específicos, y el modelo de agente que especifica los acuerdos que se dan en las interacciones entre los agentes.

En cuanto a su arquitectura, AMKM considera dos capas: una de operación y otra de facilitación. En la capa de operación se cuenta con tres tipos de agentes: los asistentes personales, los agentes de tareas y los agentes de conocimiento. En el nivel de facilitación, un grupo de agentes ayuda a los agentes a localizarse mutuamente basándose en sus necesidades; también aseguran las interacciones monitoreando las negociaciones entre agentes.

## 5.4. Trabajos futuros

El presente trabajo de tesis puede extenderse en varios sentidos. A continuación se presenta una clasificación de los trabajos futuros propuestos:

### 5.4.1. Nuevos servicios

#### *Servicio de Ontología*

El servicio de ontologías presentado en esta tesis fue simulado con ayuda de la base de datos y el Ontology Agent. Un servicio especializado en ontologías podría enriquecer el campo de aplicación de JITIK.

#### *Servicio de consultas en bibliotecas digitales*

El servicio de bridge a una biblioteca digital en esta tesis fue simulado con ayuda de la base de datos y el Bridge Agent. La notificación automática de un sistema de

biblioteca que aporte conocimiento a JITIK podría ser desarrollado.

#### *Servicio de búsqueda en Memorias Organizacionales*

Tomar ventaja del conocimiento almacenado en el ámbito de una organización en particular en particular y realizar búsquedas que ayude a aprender de experiencias pasadas, a reutilizar conocimiento y a mejorar procesos.

#### *Servicio de minería de datos*

Integrar otra técnica de Inteligencia Artificial como es la minería de datos donde se busca la relación entre grandes cantidades de información para transformarla en conocimiento.

#### *Razonamiento basado en casos*

Esta es una técnica de resolución de problemas que podría resultar de utilidad a los servicios de JITIK. Se refiere al proceso de resolver nuevos problemas basado en la resolución de problemas similares en el pasado. [19] Servicios de help-desc o diagnóstico de enfermedades son el tipo de problemas que pueden resolverse con esta técnica.

#### *Descubrimiento de Web Services*

Existen muchos servicios Web en Internet que publican información muy variada, tipo de cambio de monedas, el valor de las acciones en la bolsa, el clima, etc. Un agente inteligente que descubriera los servicios Web disponibles en Internet y que los hiciera disponible para JITIK utilizando los estándares de SOA (Service Oriented Architecture).

### **5.4.2. Mejoras a la arquitectura de JITIK**

Una mejora que podría incluirse en la Arquitectura de JITIK es la capacidad de negociación de los agentes.

Negociación es una forma de interacción que ocurre entre agentes con metas distintas. Es proceso en el cual se llega a un punto de decisión común entre dos o más agentes, cada uno tratando de alcanzar su objetivo. Los agentes primero comunican sus posiciones, las cuales pueden estar en conflicto y entonces tratan de lograr un acuerdo haciendo concesiones o buscando alternativas. [17]

Con esta capacidad JITIK podría tal vez encomendar a distintos agentes la tarea de buscar información relacionada en diferentes fuentes de información con un área de conocimiento y mediante negociación decidir cual resultado tiene más relevancia para el área.

O bien, la búsqueda de algún artículo podría ser encomendada a varios agentes en JITIK y la negociación podría estar relacionada con conseguir el mejor precio.

### **5.4.3. Relacionados con la Administración del Conocimiento**

Una actividad que resultaría útil para conocer el impacto de JITIK en AC sería medir que tanto pueden beneficiarse las actividades de distribución de conocimiento con el uso de la arquitectura de JITIK.

Para hacer esto habría que definir algunas métricas tales como productividad, calidad, facilidad de uso, etc. para luego hacer comparación de los resultados sin utilizar JITIK y posteriormente con el uso de la arquitectura. Esto de preferencia dentro de una organización.

## Capítulo 6

# Apéndice A - Pseudocódigo del prototipo de JITIK

### 6.1. Pseudocódigo del agente de sitio

```
SiteAgent(Mensaje)
<Inicio>
  <Mientras Mensaje Hacer>
    <Si Mensaje = CambioPaginaWeb Entonces>
      <Mientras PersonalAgent Hacer>
        <Si PersonalAgent = EnLinea Entonces>
          NotificarCambioPaginaWeb(Personal Agent)
        <Si No>
          GuardarMensaje(PersonalAgent)
          Siguiente PersonalAgent
        <Fin Mientras>
    <Si No Si Mensaje = MensajeArea Entonces>
      <Mientras PersonalAgent Hacer>
        <Si PersonalAgent = EnLinea Entonces>
          NotificarMensajeArea(Personal Agent)
        <Si No>
          GuardarMensaje(PersonalAgent)
          Siguiente PersonalAgent
        <Fin Mientras>
    <Mientras SiteAgent Hacer>
      <Si SiteAgent = EnLinea Entonces>
        NotificarMensajeArea(SiteAgent)
      <Si No>
        GuardarMensaje(SiteAgent)
        Siguiente SiteAgent
      <Fin Mientras>
    <Si No Si Mensaje = MensajeGrupo Entonces>
      <Mientras PersonalAgent Hacer>
        <Si Interesado(PersonalAgent,Grupo) Entonces>
          <Si PersonalAgent = EnLinea Entonces>
            NotificarMensajeArea(Personal Agent)
          <Si No>
            GuardarMensaje(PersonalAgent)
          Siguiente PersonalAgent
        <Fin Mientras>
      <Mientras SiteAgent Hacer>
        <Si SiteAgent = EnLinea Entonces>
          NotificarMensajeArea(SiteAgent)
        <Si No>
          GuardarMensaje(SiteAgent)
        Siguiente SiteAgent
      <Fin Mientras>
```

```

<Si No Si Mensaje = MensajePersonal Entonces>
  <Si PersonalAgent = Local Entonces>
    <Si PersonalAgent = EnLinea Entonces>
      NotificarMensajePersonal(Personal Agent)
    <Si No>
      GuardarMensaje(PersonalAgent)
  <Si No>
    <Mientras SiteAgent Hacer>
      <Si SiteAgent = EnLinea Entonces>
        NotificarMensajePersonal(SiteAgent)
      <Si No>
        GuardarMensaje(SiteAgent)
      Siguiente SiteAgent
    <Fin Mientras>
  Siguiente Mensaje
<Fin Mientras>

<Mientras MensajePendienteSite Hacer>
  <Si SiteAgent = EnLinea Entonces>
    NotificarMensaje(SiteAgent)
    CambiarEstatus(Mensaje,Entregado)
  <Fin Si>
  Siguiente MensajePendienteSite
<Fin Mientras>

<Mientras MensajePendienteGrupo Hacer>
  <Si SiteAgent = EnLinea Entonces>
    NotificarMensaje(SiteAgent)
    CambiarEstatus(Mensaje,Entregado)
  <Fin Si>
  Siguiente MensajePendienteGrupo
<Fin Mientras>
<Fin>

```

## 6.2. Pseudocódigo del agente de ontología

OntologyAgent(Usuario, Area)

```

<Inicio>
  Respuesta = Falso
  <Repetir>
    Leer Area
    Leer AreasxUsuario
    <Si Existe AreaxUsuario Entonces>
      Respuesta = Verdadero
    <Si no>
      Area.AreaId = Area.RelatedTo
    <Hasta Respuesta = Verdadero or Area = vacio>
    Contestar(Respuesta)
  <Fin>

```

OntologyAgent(Area)

```

<Inicio>
  Respuesta = Vacio
  <Repetir>
    Leer AreasxUsuario donde AreasxUsuario.Area = Area
    <Mientras Existe AreaxUsuario Entonces>
      Respuesta = Respuesta + Usuario
      Siguiente AreasxUsuario donde AreasxUsuario.Area = Area

```

```

    <Fin Mientras>
    Leer Area donde Area.AreaId = Area
    <Si Area.RelatedTo <> vacio Entonces>
        Area = Area.RelatedTo
        Leer Area donde Area.AreaId = Area
    <Fin Si>
    <Hasta Area = vacio>
    Contestar(Respuesta)
<Fin>

```

### 6.3. Pseudocódigo del agente personal

```

PersonalAgent()
<Inicio>
    RegistroSiteAgent()
    <Mientras MensajePendiente Hacer>
        MostrarMensaje()
        CambiarEstatus(Mensaje,Entregado)
        Siguiente MensajePendiente
    <Fin Mientras>
    <Mientras Mensajes Hacer>
        <Si Mensaje = CambioPaginaWeb Entonces>
        <Si Interesado(PaginaWeb)=Verdadero Entonces >
            MostrarMensaje()
        <Si No Si Mensaje = MensajeArea Entonces>
            <Si OntologyAgent(PersonalAgent,Area)=Verdadero Entonces >
                MostrarMensaje()
            <Si No Si Mensaje = MensajeGrupo Entonces>
                MostrarMensaje()
            <Si No Si Mensaje = MensajePersonal Entonces>
                MostrarMensaje()
        Siguiente Mensaje
    <Fin Mientras>
<Fin>

```

### 6.4. Pseudocódigo del agente monitor

```

MonitorAgent()
<Inicio>
    Leer PaginaWeb
    <Mientras PaginaWeb Hacer>
        <Si MonitorearAhora = Verdadero Entonces>
            <Si CambioPaginaWeb() = Verdadero Entonces>
                PaginaWeb.LastStatus = GuardarUltimoEstado(PaginaWeb)
                NotificarCambio(PaginaWeb)
            <Fin Si>
        <Fin Si>
    Leer Siguiente PaginaWeb
    <Fin Mientras>
<Fin>

```

## 6.5. Pseudocódigo del agente bridge o puente

```
BridgeAgent(Area)
<Inicio>
  AreasRelacionadas(Area,ListaAreas)
  <Mientras ListaAreas Hacer>
    Leer LibrosxArea donde LibrosxArea = Area
    <Mientras LibrosxArea>
      NotificarCambio(Libro)
      Leer LibrosxArea donde LibrosxArea = Area
    <Fin Mientras>
  Siguiete ListaAreas
<Fin Mientras>
<Fin>
```

## Capítulo 7

### Apéndice B - Tablas de datos para prototipo JITIK

#### 7.1. Tabla Usuarios

UserId	UserName
strujillo	Susana Trujillo
esotelo	Eduardo Sotelo
hgomez	Hugo Gómez
ccantu	Carlos Cantú
dflores	David Flores
jsanchez	José Maria Sánchez
erikab	Erika Buenrostro

#### 7.2. Tabla PáginasWeb

PageId	URL	Frequency	LastStatus	LastTimeVisit
001	<a href="http://www.websense.com/global/en/PressRoom/">http://www.websense.com/global/en/PressRoom/</a>	1440	Null	00/00/0000
002	<a href="http://trendmicro.mediaroom.com/">http://trendmicro.mediaroom.com/</a>	1440	Null	00/00/0000
003	<a href="http://www.symantec.com/about/news/release/">http://www.symantec.com/about/news/release/</a>	1440	Null	00/00/0000
004	<a href="http://www.altonivel.com.mx/newsletter.html">http://www.altonivel.com.mx/newsletter.html</a>	2880	Null	00/00/0000
005	<a href="http://www.bluecoat.com/news/pressreleases">http://www.bluecoat.com/news/pressreleases</a>	1440	Null	00/00/0000
006	<a href="http://www.ironport.com/toc/">http://www.ironport.com/toc/</a>	60	Null	00/00/0000
007	<a href="http://www.bsecure.com.mx/">http://www.bsecure.com.mx/</a>	2880	Null	00/00/0000



### 7.3. Tabla Grupos

GrupoId	GrupoName
01	Comité de Graduación MIT2008
02	Sociedad Alumnos
03	Subasta Autos a Escala

### 7.4. Tabla Areas

AreaId	AreaDescription	RelatedTo
I	Computing Methodologies	
I.2	Artificial Intelligence	I
I.2.0	General	I.2
I.2.1	Applications and Expert Knowledge-Intensive Systems	I.2
I.2.2	Automatic Programming	I.2
I.2.3	Deduction and Theorem Proving and Knowledge Processing	I.2
I.2.4	Knowledge Representation Formalisms and Methods	I.2
I.2.5	Programming Languages and Software	I.2
I.2.6	Learning	I.2
I.2.7	Natural Language Processing	I.2
I.2.8	Problem Solving, Control Methods, and Search	I.2
I.2.9	Robotics	I.2
I.2.10	Vision and Scene Understanding	I.2
I.2.11	Distributed Artificial Intelligence	I.2
I.2.12	Intelligent Web Services and Semantic Web	I.2
I.2.13	Knowledge Management	I.2
I.2.m	Miscellaneous	I.2

### 7.5. Tabla LibrosxArea

Classification	Title	Author	Editor	AreaId
QA76.76.I58 A72 2007	Architectural design of multi-agent systems : technologies and techniques	1Hong Lin	Hershey	I.2.11

## 7.6. PaginasWebxUsuario

UserId	PageId
dflores	001
dflores	005
dflores	002
ccantu	004
strujillo	007
esotelo	004
erikab	005

## 7.7. Tabla GruposxUsuario

UserId	GrupoId
dflores	03
dflores	02
strujillo	01
ccantu	02
hgomez	03
esotelo	01
erikab	02

## 7.8. Tabla AreasxUsuario

UserId	AreaId
dflores	I.2
ccantu	I.2
hgomez	I
strujillo	I
erikab	I.2

## Capítulo 8

# Apéndice C - Taxonomía de ACM para la Inteligencia Artificial

- 9. Computing Methodologies
  - 2. Artificial Intelligence
    - 0. General
      - 1. Cognitive simulation
      - 2. Philosophical foundations
    - 1. Applications and Expert Knowledge-Intensive Systems
      - 1. Cartography
      - 2. Computer vision
      - 3. Decision support
      - 4. Education
      - 5. Environment
      - 6. Games and infotainment
      - 7. Industrial automation
      - 8. Law
      - 9. Mathematics
      - 10. Medicine and science
      - 11. Military
      - 12. Natural language interfaces
      - 13. Office automation
      - 14. Space
      - 15. Transportation
    - 2. Automatic Programming
      - 1. Automatic analysis of algorithms
      - 2. Program modification
      - 3. Program synthesis
      - 4. Program transformation
      - 5. Program verification
    - 3. Deduction and Theorem Proving and Knowledge Processing
      - 1. Answer/reason extraction
      - 2. Constraint-based processing
      - 3. Deduction
      - 4. Inference engines
      - 5. Logic processing
      - 6. Logic programming
      - 7. Mathematical induction
      - 8. Metatheory
      - 9. Nonmonotonic reasoning and belief revision
      - 10. Resolution
      - 11. Rule-based processing
      - 12. Uncertainty, "fuzzy," and probabilistic reasoning
    - 4. Knowledge Representation Formalisms and Methods
      - 1. Agent communication languages
      - 2. Distributed representations

- 3. Frames and scripts
- 4. Knowledge base management
- 5. Knowledge base verification
- 6. Modal logic
- 7. Predicate logic
- 8. Relation systems
- 9. Representation languages
- 10. Representations (procedural and rule-based)
- 11. Semantic networks
- 12. Storage mechanisms
- 13. Temporal logic
- 5. Programming Languages and Software
  - 1. Expert and knowledge-intensive system tools and techniques
- 6. Learning
  - 1. Analogies
  - 2. Concept learning
  - 3. Connectionism and neural nets
  - 4. Heuristics design
  - 5. Induction
  - 6. Knowledge acquisition
  - 7. Machine learning
  - 8. Language acquisition
  - 9. Parameter learning
- 7. Natural Language Processing
  - 1. Discourse
  - 2. Language generation
  - 3. Language models
  - 4. Language parsing and understanding
  - 5. Language summarization
  - 6. Machine translation
  - 7. Speech recognition and synthesis
  - 8. Text analysis
  - 9. Web text analysis
- 8. Problem Solving, Control Methods, and Search
  - 1. Backtracking
  - 2. Constraint satisfaction
  - 3. Control theory
  - 4. Dynamic programming
  - 5. Graph and tree search strategies
  - 6. Heuristic methods
  - 7. Plan execution, formation, and generation
  - 8. Scheduling
- 9. Robotics
  - 1. Autonomous vehicles
  - 2. Biorobotics
  - 3. Commercial robots and applications
  - 4. Kinematics and dynamics
  - 5. Manipulators
  - 6. Nanorobots
  - 7. Neuromorphic computing
  - 8. Operator interfaces
  - 9. Propelling mechanisms
  - 10. Sensors
  - 11. Workcell organization and planning
  - 12. Vision
- 10. Vision and Scene Understanding
  - 1. 3D/stereo scene analysis
  - 2. Architecture and control structures
  - 3. Intensity, color, photometry, and thresholding
  - 4. Modeling and recovery of physical attributes

- 5. Motion
- 6. Perceptual reasoning
- 7. Representations, data structures, and transforms
- 8. Shape
- 9. Texture
- 10. Video analysis
- 11. Distributed Artificial Intelligence
  - 1. Coherence and coordination
  - 2. Intelligent agents
  - 3. Languages and structures
  - 4. Multiagent systems
- 12. Intelligent Web Services and Semantic Web
  - 1. Intelligent Web service languages
  - 2. Internet reasoning services
  - 3. Ontology design
  - 4. Ontology languages
- 13. Knowledge Management
  - 1. Knowledge acquisition
  - 2. Knowledge engineering methodologies
  - 3. Knowledge life cycles
  - 4. Knowledge maintenance
  - 5. Knowledge modeling
  - 6. Knowledge personalization and customization
  - 7. Knowledge publishing
  - 8. Knowledge retrieval
  - 9. Knowledge reuse
  - 10. Knowledge valuation
- m. Miscellaneous
  - 1. Adaptive hypermedia
  - 2. Computational neuroscience
  - 3. Evolutionary computing and genetic algorithms
  - 4. Wearable AI

## Bibliografía

- [1] Uwe M. Borghoff, Remo Pareschi, *Information Technology for Knowledge Management*, Ed. Springer, 1998.
- [2] Gregoris Mentzas, Dimitris Apostolou, Andreas Abecker, Ron Young, *Knowledge Asset Management*, Ed. Springer, 2003.
- [3] Amidon, Debra, *El Momentum de la Administración del Conocimiento*, [http://www.entovation.com/momentum/mom\\_es.htm](http://www.entovation.com/momentum/mom_es.htm), 2001.
- [4] James W. Cortada, John A. Woods, *The Knowledge Manager Yearbook, 1999-2000*, Ed. Elsevier, 1999.
- [5] Spek, Rob van der ; Spijkervet, André, *Knowledge Management: Dealing Intelligently with Knowledge, knowledge Management and its Integrative Elements*, Edited by Jay Liebowitz and Lyle C. Wilcox, CRC Press, 1997.
- [6] Shoham, Yoav, *An Overview of Agent-Oriented Programming Software Agents*, Edited by Jeffrey M. Bradshaw, 1997.
- [7] Manuel Arbolí, *El Negocio es el conocimiento*, Revista Alto Nivel, Num. 230, 3 de octubre de 2007.
- [8] Jay Liebowitz, *Knowledge Management, Learning from Knowledge Engineering*, CRC Press, 2001.
- [9] Gabriel Valerio, Pablo Ramírez, *El Desarrollo de la memoria organizacional a través del capital instrumental*, Transferencia Año 18, No. 71, julio de 2005.
- [10] Consorcio World Wide Web, Copyright 1994-2008 <http://www.w3c.es>.
- [11] Dan Remenyi, *Third European Conference on Knowledge Management: Trinity College Dublin.*, Management Center International. 2002. Publicado por Academic Conferences Limited.
- [12] Michael Wooldridge, *An Introduction to MultiAgents Systems*, Chichester, England : J. Wiley, c2002.

- [13] Junnarkar, Bipin, Leveraging Collective Intellect by Building Organizational Capabilities *Expert systems with Applicationst*, Special Issue: Knowledge Management, Vol. 13, Number 1, July 1997.
- [14] José Luis Aguirre, Ramón Brena, *Redes Informáticas de conocimiento y administración del conocimiento*, Documento interno de trabajo, Centro de Sistemas Inteligentes. ITESM Monterrey.
- [15] Fabian Gandon, *Distributed Artificial Intelligence and Knowledge Management: Ontologies and Multi-agent Systems for a Corporate SemanticWeb*, Scientific Philosopher Doctorate Thesis in Informatics, INRIA and University of Nice, 2002.
- [16] Qui-Jian Sheng, Zhi-Kun Zhao, Shao-Hui Liu and Zhong-Zhi Shi, A Teamwork Protocol for Multi-agent System, *Intelligent Agents and Multi-Agent Systems, 6th Pacific Rim International Workshop on Muti-Agents, PRIMA 2003.*, Institute of computing Technology, Chinese Academy of Sciences, 2003.
- [17] Gerhard Weiss, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, Edited by Gerhard Weiss, The MIT Press, 1999.
- [18] Copyright 2007, by the Association for Computing Machinery, Inc. <http://www.acm.org/class/1998/ccs98.html>.
- [19] Copyrights <http://en.wikipedia.org>.
- [20] M. Lacher, M. Koch, *An agent based knowledge management framework*, AAAI Spring Symposium 2000, 2000.
- [21] H. Wang, D. Xu, *Knowledge management systems with intelligent agent support*, [http://140.123.175.201/NetEconomy/KM/ppt/knowledge management systems with intelligent agent support.pdf](http://140.123.175.201/NetEconomy/KM/ppt/knowledge%20management%20systems%20with%20intelligent%20agent%20support.pdf).
- [22] F. Bergenti, A. Poggi, and G. Rimassa, Agent architectures and interaction protocols for corporate management systems, *ECAI Workshop on Knowledge Management and Organisational Memories* , Berlin Germany,2000.
- [23] J. Chen, S. Wolfe, and S. Wragg, A distributed multi-agent system for collaborative information management and sharing, *9th ACM International Conference on Information and Knowledge Management* , ACM,2000.
- [24] L. van Elst, A. Abecker, Domain ontology agents in distributed organizational memories, *IJCAI'2001 Workshop on Knowledge Management and Organisational Memories* , Seattle Washington,2001.

- [25] A. Preece, K. Hui, W. Gray, P. Marti, T. Bench-Capon, D. Jones, Z. Cui, The KRAFT architecture for knowledge fusion and transformation, *19th SGES International Conference on Knowledge-based Systems and Applied Artificial Intelligence*, 1999.
- [26] M. Koch, Knowledge management and knowledge agents in Campiello, *Workshop on Intelligent Agents in CSCW*, , Editors B. Lees, H. Muller, C. Branki, Dortmund, Germany, 1998.
- [27] V. Dignum, *An overview of agents in knowledge management*, <http://www.cs.uu.nl/virginia/amkm.pdf>.
- [28] Copyrights *jade.tilab.com*.
- [29] Copyrights <http://labs.bt.com/projects/agents/zeus>.



